MODELLING OF CO_2 STORAGE IN NATURALLY FRACTURED RESERVOIRS

RAFAEL MARCH CASTANEDA NETO

Submitted for the degree of Doctor of Philosophy
Institute of Petroleum Engineering
Heriot-Watt University
May 2018

ABSTRACT

Geological storage of CO_2 is regarded as an important technology to achieve the targets of temperature increase established in the last decade. Naturally fractured reservoirs (NFRs) are ubiquitous across the world, but so far have received little attention as potential CO_2 storage sites. Among the main concerns regarding storage in NFRs are the fast flow of the CO_2 plume in the fracture network and the high capillary contrast between fractures and matrix that may keep CO_2 in the fractures, which have very low pore volume. This thesis provides a systematic study of CO_2 storage in NFRs. The dual-porosity method, which is typically used for simulating flow in NFRs, is used to evaluate under what geological conditions fractured formations can be used for storage. The dual-porosity method relies on transfer functions that model the matrix-fracture fluid exchange. New transfer functions that capture the transfer physics more accurately than existing models available in the literature are developed. These transfer functions are then applied to understand storage capacity and CO_2 plume flow for conceptual fractured aquifers. The results presented in this thesis show that in principle fractured formations have storage potential if the geological conditions are suitable and injection rates are managed appropriately.

To my family.

ACKNOWLEDGMENTS

My PhD was made possible through financial grants provided by the Department of Energy of the United States of America and by the Energi Simulation Foundation (formerly known as Foundation CMG). I am very grateful to these organizations and to the persons who made these grants possible.

My family was important in every moment of my life, and it was not different during my PhD. I would like to thank them for the extremely valuable support they have provided me not only during the development of this thesis, but also during my entire life.

I also would like to acknowledge the importance of two persons in the achievement of this thesis: Florian Doster and Sebastian Geiger (order does not matter). My supervisors have invested in my research career and have believed in my potential. They have pushed me up when I was a bit unmotivated and have challenged me when I was too confident. They have taught me how to be a scientist, which is a dream of mine since I was seventeen years old. Everything that I accomplish from now on is a direct result from the support I got from them, and I will be forever thankful for this.

This journey would not have been so smooth without the company of my friends. I thank initially the first friends I had in Edinburgh: Daniel Schulte, Alessandro Mangione and Christine Maier. They made me feel welcome since my very first day. A big thank you also to the friends that came a bit later, most of them part of the Carbonates group: Simeon, Zhao, Tianshen, David, Masoud, Daniel, Ali, Jackson, Emmanuel, Herwald, Jorge, Saeeda, Fran, Ilaria, Jane, Fahed, Mohamed, Saideep, Victoria, Zainab, Rhona, Lesly, Julien, among others. You all make the grey-looking Energy Academy building much more colourful !

To the Brazilian and Portuguese community, many of them not in Edinburgh anymore: it was great to speak some Portuguese with you. Special thanks goes to: Nathalia, Odilla, Samantha, Geraldo, Juliana, Sandra, Suzanny, Andre, Murillo, Hannah, Duarte, Gustavo, Hydra, Gabriel, Cayo, Joao, Diogo and Goncalo. For those of you who left, hopefully we meet again in Portugal, Brazil or elsewhere.

During my PhD I had several valuable experiences abroad. Thanks to Professor Mike Celia, Dr. Bo Guo and Dr. Karl Bandilla from Princeton University for the incredibly valuable discussions during our meetings in the US; which always made me see things in a different perspective. I spent a month as a Visiting Research Scholar at Penn-State University, where I could learn a lot and met great

scientists and persons. I thank Professor Luis Ayala, Dr. Miao Zhang and Giancarlo Bonotto, who is now a personal friend. It was great spending that month with you. I also spent the summer of 2016 at Aramco Services Company, TX, US as Research Intern. A special thanks to Susan Agar for this opportunity and to the rest of the Geoscience Technology Team for the fruitful discussions and tasty lunches. A special acknowledgement here goes to Yan Zaretskiy, who quickly became a friend of mine and with whom I have learned a lot about different things; most of them related to numerics and Python programming.

Last but not least, I thank the Scots for being such a warm-hearted and welcoming people ! There was no single moment when I thought I was not welcomed here in Edinburgh. From the shop cashiers to every local I happened to meet here: keep it up in making this a perfect environment for students coming from abroad.

Edinburgh, May 2018

ACADEMIC REGISTRY

Research Thesis Submission

Name*:*	RAFAEL MARCH CASTANEDA NETO		
School:	EGIS – INSTITUTE OF PETROLEUM ENGINEERING		
Version: *(i.e. First, Resubmission, Final)*	FINAL	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

CONTENTS

ACRONYMS

AD - Automatic-Differentiation (Framework)

DP - Dual-Porosity

GD - Gravity Drainage

MRST - Matlab Reservoir Simulation Toolbox

NFR's - Naturally Fractured Reservoirs

SI - Spontaneous Imbibition

TF - Transfer Function

VI - Vertically-Integrated (Model)

VIDP - Vertically-Integrated Dual-Porosity (Model)

PUBLICATIONS

This thesis contains excerpts from the following papers:

March, R., Doster, F., Geiger, S. (2016). Accurate early-time and late-time modeling of countercurrent spontaneous imbibition. Water Resources Research, 52, 6263-6276, https://doi.org/10.1002/2015WR018456.

March, R., Elder, H., Doster, F., Geiger, S. (2017). Accurate dual-porosity modeling of CO_2 storage in fractured reservoirs (SPE-182646-MS). SPE Reservoir Simulation Conference, Montgomery, USA, 20-22 February.

March, R., Doster, F., Geiger, S. (2018). Assessment of CO_2 Storage Potential in Naturally Fractured Reservoirs With Dual-Porosity Models. Water Resources Research, 54, 1-19, https://doi.org/10.1002/2017WR022159.

The following paper was published during this thesis and helped in its the development:

Egya, D. O., Geiger, S., Corbett, P. W. M., **March, R.**, Bisdom, K., Bertotti, G., Bezerra, F.H. (2018). Analysing the limitations of the dual-porosity response during well tests in naturally fractured reservoirs. Petroleum Geoscience, https://doi.org/10.1144/petgeo2017-053

1

INTRODUCTION AND BACKGROUND

Concern about the rising levels of anthropogenic CO_2 in the atmosphere is growing, as CO_2 is one of the major contributors to the greenhouse effect. During the past two decades, the identification and development of reliable and effective techniques to mitigate the environmental effects of greenhouse gases have received considerable attention in the scientific research community. These techniques include Carbon Capture and Utilization (CCU), Negative Emission Technology (NET) and Carbon Capture and Storage (CCS)
(Cuellar-Franca and Azapagic, 2015; IPCC, 2005; Ringrose, 2017), among others.

Geological storage of CO_2 is regarded as an important technology to achieve the targets of temperature increase established in the last decade and updated during the Paris climate agreement negotiated in December 2015. In Pacala and Socolow (2004), the authors have argued that a mix of different technologies, including CCS, will have to be employed in order to slow down climate change over the next five decades. They estimated that 3500 projects of the size of the *Sleipner* CCS project in the North Sea would be required to address one seventh of the global temperature increase issue. In Celia et al. (2015), the authors later updated these numbers, using present-day data, and calculated that this number of projects would solve just one tenth of it. Therefore, the number of CCS projects needs to increase considerably to have a noticeable impact on the environment. At this scale, the availability of geological formations suitable for CO_2 injection at locations that allow economic operations is a limiting factor that hinders the global implementation of carbon storage.

Fractured reservoirs are ubiquitous in sedimentary basins across the world. Estimates suggest that more than half of the world's remaining conventional hydrocarbon resources are stored in extensively fractured carbonate reservoirs (Burchette, 2012). However, these reservoirs have received little attention as potential candidates for CO_2 storage. Yet, they offer a significant potential for combining CO_2 storage with enhanced oil recovery (Agada et al., 2016). The primary reason seems to be related to the fact that for most formations, CO_2 is the non-wetting phase relative to the resident fluid. Since capillary pressure in the fractures is typically orders of magnitude lower than in the rock matrix, capil-

lary forces may prevent CO_2 from invading the rock matrix. This could severely impact the storage potential of the aquifer, as the matrix generally accounts for most of the pore volume. This negative effect is more pronounced in aquifers with small thickness (< 50 m) or with a set of near-horizontal fractures with apertures that are large enough to prevent the creation of capillary bridges across the vertical section and hence there is no capillary continuity (Aspenes et al., 2008; Labastie, 1990). Fractures also have high conductivity and the presence of an interconnected system of fractures may lead to fast flow of CO_2 through the fracture system, without significant storage in the rock matrix. This could lead to a loss in control of the CO_2 plume, and potentially to CO_2 leakage through abandoned wells or reactivated faults. Despite these challenges, the study of CO_2 storage in fractured formations is very relevant to the global implementation of CCS. Firstly, fractures are a major source of uncertainty in geological formations (Narr et al., 2006). Hence, the existence of unidentified fracture networks should be incorporated in the risk analysis of storage projects. Secondly, as fractured reservoirs are numerous, they may be considered as storage sites as the number of CCS projects increases. The ability to model and simulate the multi-phase flow dynamics in such formations is of utmost importance to the development of a storage project.

This thesis focuses on modelling and simulation of CO_2 storage in naturally fractured reservoirs (NFR's). We begin this introductory chapter by discussing the main physical mechanisms that take place during CO_2 injection in NFR's. We then present the governing equations and numerical methods that allow the simulation of multiphase flow in fractured formations. Finally, we present a detailed summary of the objectives of this thesis as well as its structure.

1.1 CO2 STORAGE DYNAMICS IN NFR'S

We start by reviewing the main physical mechanisms that take place during CO_2 injection in fractured formations. In this thesis, we focus our discussion on fractured saline aquifers, because saline aquifers have the largest storage potential (IPCC, 2005). CO_2 storage operations target geological formations whose temperature and pressure conditions are such that CO_2 is in supercritical state. In this state, the gas density increases significantly, which improves its stored mass per unit pore volume. Typical values for CO_2 density range from 266 to 733 kg/m^3, depending on the pressure and temperature conditions of the formation. The compressibility, on the other hand, ranges typically from 10^{-9} to 10^{-8} Pa^{-1}, which is at least one order of magnitudes lower than that of a gas (Law and Bachu, 1996). Brine density is most sensitive to the salinity of the water, with values ranging from 945 to 1230 kg/m^3 (Nordbotten and Celia, 2012); its com-

pressibility is of the order of $10^{-10}\ Pa^{-1}$. The density of brine is therefore always larger than the density of CO_2. This density difference induces upward flow of CO_2. Therefore, suitable geological formations for storage should be overlaid by a capping formation that prevents CO_2 from rising to the surface. CO_2 will also dissolve in the resident brine, with typical values of solubility by mass ranging from 1% to 6% (Spycher et al., 2003). The dissolution of CO_2 leads to an increase in the density of brine, which may create an unstable system and therefore the generation of convective dissolution cells. The convective dissolution of CO_2 in brine has been identified as an important mechanism for safe storage of CO_2 in unfractured saline aquifers (Emami-Meybodi and Hassanzadeh, 2015; Macminn et al., 2011; Riaz et al., 2006; Suekane et al., 2008). In summary, the injection of CO_2 in a saline aquifer leads to a multi-phase multi-component system, with distinct time-scales associated with buoyancy, injection and mass transfer between the phases (see Figure 1.1).

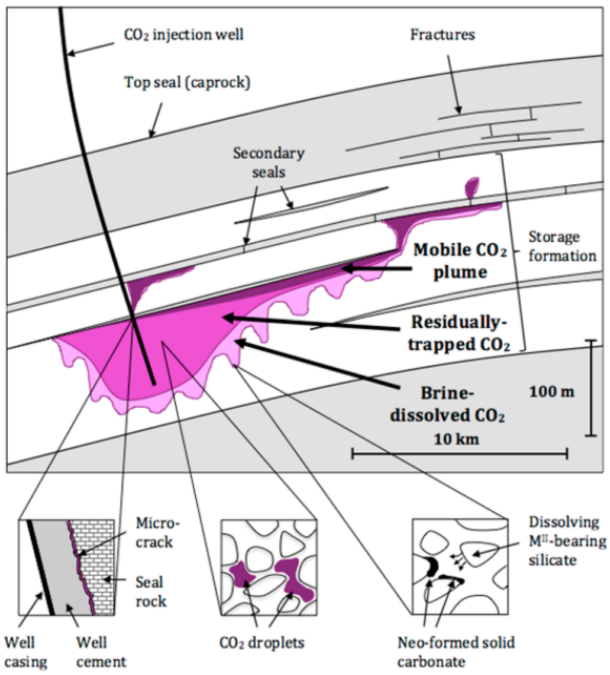

Figure 1.1: Schematic figure illustrating the distribution of CO_2 near an injection well roughly one decade after the end of injection and the four trapping mechanisms that enable geological carbon sequestration (figure taken from Bourg et al. (2015)).

Modelling of fractured formations faces an additional challenge as fractures introduce additional time-scales to the system. An inter-connected system of fractures can act as a high-permeability pathway that enables rapid fluid flow. Hence, injected CO_2 can quickly travel through the fracture system, while its transfer to the rock matrix will occur at at a time-scale that is usually slower (Figure 1.2(a)). Fracture-matrix transfer mechanisms were extensively studied through laboratory and numerical experiments (e.g., (Ahmed Elfeel et al., 2016;

Fernø et al., 2013; Lemonnier and Bourbiaux, 2010; Lu et al., 2008)). Figure 1.2(b) summarizes the most relevant transfer mechanisms. If the permeabilities of fracture and matrix are comparable or if there is capillary continuity between the matrix blocks, CO_2 enters the rock matrix blocks by viscous displacement. Otherwise, fractures will quickly transport the CO_2 phase, creating isolated systems of brine-saturated matrix blocks surrounded by CO_2 saturated fractures. Each matrix block is not in equilibrium in this configuration, as brine is heavier than CO_2 and will redistribute with the supercritical fluid, during a drainage process driven by capillary and gravity forces. The mechanism is illustrated as "buoyancy redistribution" in Figure 1.2(b). When the injection rate is reduced or stopped, some of the CO_2-saturated fractures get filled by water again. Capillary forces then cause the wetting phase to imbibe the rock matrix, displacing CO_2 back to the fractures (March et al., 2016). Figure 1.2(b) illustrates this process as "capillary imbibition". Not all the CO_2 flows back into the fractures but remains trapped in the pores of the rock due to the hysteretic nature of the relative permeability (Juanes et al., 2006).

We assume these two mechanisms - gravity drainage during CO_2 injection phase followed by capillary imbibition during plume migration when injection stops - are the main mechanisms responsible for fluid transfer between fractures and matrix during CO_2 storage operations in NFRs during the first hundreds of years of operation. Other mechanisms like viscous transfer, fluid compression and dissolution are unlikely to contribute to the transfer of CO_2 to the matrix at the timescales considered here. Viscous transfer may be relevant in the near-well region, where large pressure gradients induced by the well may generate a horizontal component inside the matrix blocks. We consider in this thesis NFRs with an interconnected system of fractures and good permeability contrast between fracture and matrix. Therefore, far from the near-well region, pressure gradients experienced by the formation will generally lead to a fast flow of CO_2 in the fractures, with a negligible amount of CO_2 invading the matrix due to viscous displacement. While it is generally accepted that a "good" permeability contrast between fractures and matrix allows us to neglect the viscous transfer, the literature lacks a systematic investigation of how high this contrast should be. A conservative estimate of the ratio between flow in matrix and fracture (denoted here as Q_m and Q_f) may be carried out by considering both continua as a layered system consisting of two layers (fracture system and matrix) with different

thickness and permeabilities. The (single-phase) flow in each layer is then given by Darcy's law:

$$Q_m = \frac{k_m w h_m \Delta p}{\mu L}, \tag{1.1}$$

$$Q_f = \frac{k_f w h_f \Delta p}{\mu L}, \tag{1.2}$$

where Δp is the applied pressure gradient, k_i is the permeability of layer (medium) i, w the width of the media, h_i the height of of layer i, μ the fluid viscosity and L the system's length. If we consider that the height of each layer scales with the porosity of the media ($h_i = \phi_i H$, where H is the height of the layered system), we obtain $Q_m/Q_f = \left(\phi_m/\phi_f\right)\left(k_m/k_f\right)$. According to this estimate, a porosity ratio of $\phi_m/\phi_f = 10$ and a permeability ratio of $k_m/k_f = 10^{-3}$ would lead to $Q_m/Q_f = 10^{-2}$, implying that the amount of flow that goes to the matrix is negligible.

CO_2 dissolution in brine happens at timescales of hundreds of years, much slower than the typical timescales of CO_2 injection considered here. Convection-driven dissolution ("density fingers") is regarded as a key physical mechanism to shorten this timescale (Riaz et al., 2006). The high-permeability fractures, however, represent a preferential pathway for the development of convection cells. This hinders the development of convection cells within the matrix blocks (Vujevic et al., 2014). Finally, the low compressibility of the resident brine allows us to neglect the transfer due to brine compression during CO_2 injection. To conclude, we present a quick and conservative estimate of the timescale of pressure equilibration inside a matrix block. Considering a permeability of $k_m = 1$ mD, a porosity of $\phi_m = 0.1$, a brine compressibility of $c = 10^{-10}\ Pa^{-1}$, a viscosity of $\mu_w = 1$ cP and a characteristic block dimension of $L = 100$ m, the pressure diffusion timescale is estimated by $T \approx L^2/\eta$, with $\eta = k_m/\phi_m \mu_w c$. This yields an equilibration time of ≈ 1 day. Equilibration happens instantaneously within the operation's timescale, on the order of years.

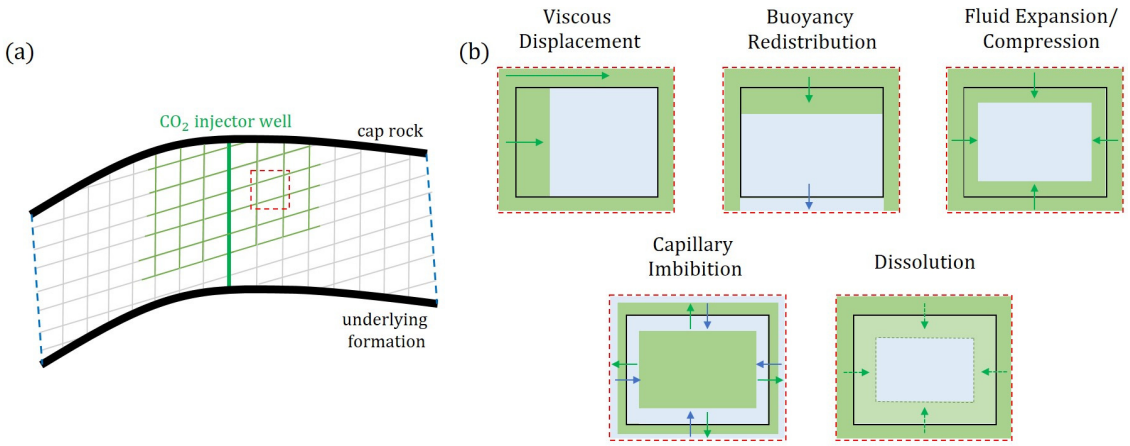

Figure 1.2: (a) Conceptual representation of CO_2 injection in a fractured reservoir. Injected CO_2 (in green color) quickly saturates the fractures (light-gray lines), that generally have low pore volume and high permeability. (b) Conceptualization of different transfer mechanisms responsible for transfer of CO_2 (green) and brine (blue) between a matrix block and the surrounding fractures. Geomechanical effects in the transfer are neglected here, but may be important for some formations. Arrows indicate the main direction of CO_2 and brine flow on each mechanism (green and blue, respectively). The dominant mechanisms in a well-connected fracture network are buoyancy redistribution and capillary imbibition.

1.2 GOVERNING EQUATIONS FOR CO2-BRINE FLOW IN POROUS MEDIA

We introduce the system of conservation equations for the wetting (brine) and non-wetting phases (CO_2) that allow us to model multiphase flow in porous media. Unless stated otherwise, in the remainder of this thesis we consider the following simplifying assumptions:

- **Geomechanical effects are neglected.** While we acknowledge the importance of geomechanics in modelling of CO_2 storage, we disregard geomechanical effects in this thesis. We follow the conventional approach, where Darcy's law is used to capture the momentum transfer between the fluid and the porous medium. In a real fractured aquifer where CO_2 is being injected, fractures may close/open based on the stress state. This effect causes changes in flow rates in fractures and matrix and may turn a dual-porosity system into a single-porosity. In this thesis, we assume that the dual-porosity model holds. Geomechanics may also induce inter-porosity fluid exchange. For instance, brine may be transferred to the fractures due to block shrinkage. However, we do not believe that this invalidates the facts that 1) capillary pressure must be overcome to allow CO_2 to saturate the matrix; and 2) the main driving force behind the drainage of the blocks is created by the buoyancy column in the fractures; which are the main physical mechanisms that receive focus in this thesis.

- **Phases are immiscible and geochemical reactions are neglected.** It is observed that carbon dioxide in the subsurface can undergo a sequence of geochemical interactions with the rock and formation water (IPCC, 2005). CO_2 dissolves in the formation water and may react with the formation minerals, creating or destroying porosity via rock dissolution and mineral precipitation. Mineral trapping is considered the safest storage mechanism that may securely arrest CO_2 for thousands of years. However, it is also believed to be comparatively slow, taking a thousand years or longer, due to low reaction rates. Solubility of CO_2 in brine increases with decreasing temperature and salinity. As shown in the Chapters 3 and 5, these are environments that are not favourable for CO_2 storage in the context of fractured formations, as the decreased buoyancy drive may not allow CO_2 to overcome capillary forces. In this situation, dissolution of CO_2 in the brine-saturated matrix blocks could be an important mechanism to trap CO_2 in the matrix. Despite the importance of this trapping mechanism, in this thesis we focus in the hydrodynamical phenomena associated with CO_2/brine flow in fractured formations, leaving the investigation of the chemical effects for a future work.

- **Phases are incompressible.** We assume that phases are incompressible, with average densities determined by the average pressure and temperature conditions of the formation. This simplification is motivated by the fact that compressibility of brine and CO_2 in its supercritical state are low. The dissolution of CO_2 in resident brine of the matrix blocks could increase the compressibility of the phase. However, as stated above, CO_2 concentrations are low in the matrix blocks during the injection phase, since convection cells that enhance dissolution will form in the fractures. Regarding the effect of compressibility in the CO_2 plume in the fractures; we refer to Vilarrasa et al. (2010), who observe that the error in the interface position caused by neglecting CO_2 compressibility is relatively small when viscous forces dominate. We believe we are in the viscous regime, since we are considering the injection period.

The following set of partial differential equations describes the flow of immiscible fluids in porous media (Bear, 1972):

$$\begin{aligned} &\frac{\partial}{\partial t}\left(\phi \rho_\alpha S_\alpha\right) + \nabla \cdot \rho_\alpha \mathbf{q}_\alpha = \tilde{q}_\alpha, \\ &\mathbf{q}_\alpha = -\mathbf{k}\lambda_\alpha \left(\nabla p_\alpha - \rho_\alpha \mathbf{g}\right), \end{aligned} \tag{1.3}$$

where S_α is the saturation of phase α, ρ_α its density, $\tilde{q}_\alpha$ mass source terms (e.g. due to wells) and ϕ and $\mathbf{k}$ the porosity and permeability of the porous medium. The multiphase Darcy model relates the Darcy velocity $\mathbf{q}_\alpha$ of phase α with the

gradient of the phase's pressure p_α and its mobility, defined as $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$, where μ_α is its viscosity and $k_{r\alpha}$ is the relative permeability to phase α. Phase pressures are related by the capillary pressure through the expression $p_n = p_w + p_c(S_w)$. The incompressibility assumption allows us to extract the density from the derivatives, reducing the system defined by Equations 1.3 to:

$$\begin{aligned} &\phi\rho_\alpha \frac{\partial S_\alpha}{\partial t} + \rho_\alpha \nabla \cdot \mathbf{q}_\alpha = \tilde{q}_\alpha, \\ &\mathbf{q}_\alpha = -\mathbf{k}\lambda_\alpha \left(\nabla p_\alpha - \rho_\alpha \mathbf{g}\right). \end{aligned} \tag{1.4}$$

We assume Corey-type relationships for the relative permeability and capillary pressure. Therefore, we write

$$\begin{aligned} &k_{r\alpha} = k_{r\alpha}^{max} S_{\alpha e}^{n_\alpha}, \\ &p_c(S_w) = P_e S_{we}^{-n_p}, \end{aligned} \tag{1.5}$$

where $S_{\alpha e}$ is the effective saturation, normalized by the residual saturations, and n_α and n_p are the power-law exponents of the relative permeabilities and capillary pressure. $S_{\alpha e}$ is defined as

$$S_{\alpha e} = \frac{S_\alpha - S_{\alpha r}}{1 - \sum_\alpha S_{\alpha r}}. \tag{1.6}$$

The conservation equations defined by Equation (1.3) are identically represented by a single equation for the saturation of the wetting phase and an equation for the total flux $\mathbf{q}_t \equiv \mathbf{q}_w + \mathbf{q}_n$:

$$\begin{aligned} &\nabla \cdot \mathbf{q}_t = \tilde{q}_w/\rho_w + \tilde{q}_n/\rho_n, \\ &\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \mathbf{q}_w = \tilde{q}_w/\rho_w, \\ &\mathbf{q}_w = f_w \mathbf{q}_t + k f_w \lambda_n \frac{dp_c}{dS_b} \nabla S_b + (\rho_w - \rho_n) g \nabla z, \end{aligned} \tag{1.7}$$

where $f_\alpha = \lambda_\alpha/\lambda_t$ is the fractional flow of phase α and λ_t the total flow mobility defined as $\lambda_t = \lambda_w + \lambda_n$. Equations (1.7) are usually named the *fractional flow formulation*. The formulation will be used to develop transfer functions that are based on analytical solutions in Chapters 3 and 4.

1.3 SIMULATION OF FLOW IN FRACTURED FORMATIONS

Techniques to simulate flow in fractured porous rocks are typically separated in two large families of methods. The first family include methodologies that explicitly represent the fractures as geometrical features in the simulation model.

Pertaining to this family are the Discrete Fracture and Matrix (DFM) methods (Huber and R., 1999; Hoteit and Firoozabadi, 2005; Geiger et al., 2009; Karimi-Fard et al., 2004; Moinfar et al., 2013; Schmid and Geiger, 2013) and the Embedded Discrete Fracture Model (EDFM) (Lee et al., 2001; Fumagalli et al., 2016; Shah et al., 2016; Moinfar et al., 2014). The second family of methods considers the fractures as a second continuum. Of particular importance are the Dual Porosity (Warren and Root, 1963), the Dual-Porosity Dual-Permeability (Kazemi et al., 1976), together with their extensions like the Multiple Interacting Continua (Pruess, 1985) and the Multi-Rate Dual-Porosity methods (Di Donato et al., 2007; Geiger et al., 2013; Maier and Geiger, 2013; Tecklenburg et al., 2016) and the Multiple Subregions Method (Karimi-Fard et al., 2006; Gong et al., 2008; Hui et al., 2018).

Perhaps the most used variation of the DFM method is the one introduced in Karimi-Fard et al. (2004). In this work, the author propose representing each fracture explicitly as “virtual” cells, that is, cells that are present in the final linear system but are not explicitly present in the computational mesh. This is done by considering fractures as lower-dimension elements that match the faces (or edges, for 2D meshes) of the mesh elements. Each face of the mesh that is “marked" as a fracture will hold additional degrees of freedom to represent fields that are stored in the virtual fracture cell. Intersections between multiple fractures are treated via the “star-delta” transformation that computes the transmissibility between the different fracture cells without the need of explicitly accounting for a small element in the intersection. While the DFM method is useful for applications (e.g. upscaling of fracture properties) conforming the computational mesh to a complex system of fractures can be challenging, time-consuming or even computationally unfeasible. Figure 1.3 shows a 2D triangular mesh created using a fracture mesher developed in the course of this thesis. While triangular meshes offer good flexibility to discretize complex geometries, we note in the detailed zoom of the mesh that small angles between fractures lead to very small triangles with high aspect ratio. This leads to ill-conditioned numerical systems and poses a numerical challenge to the solution of flow equations.

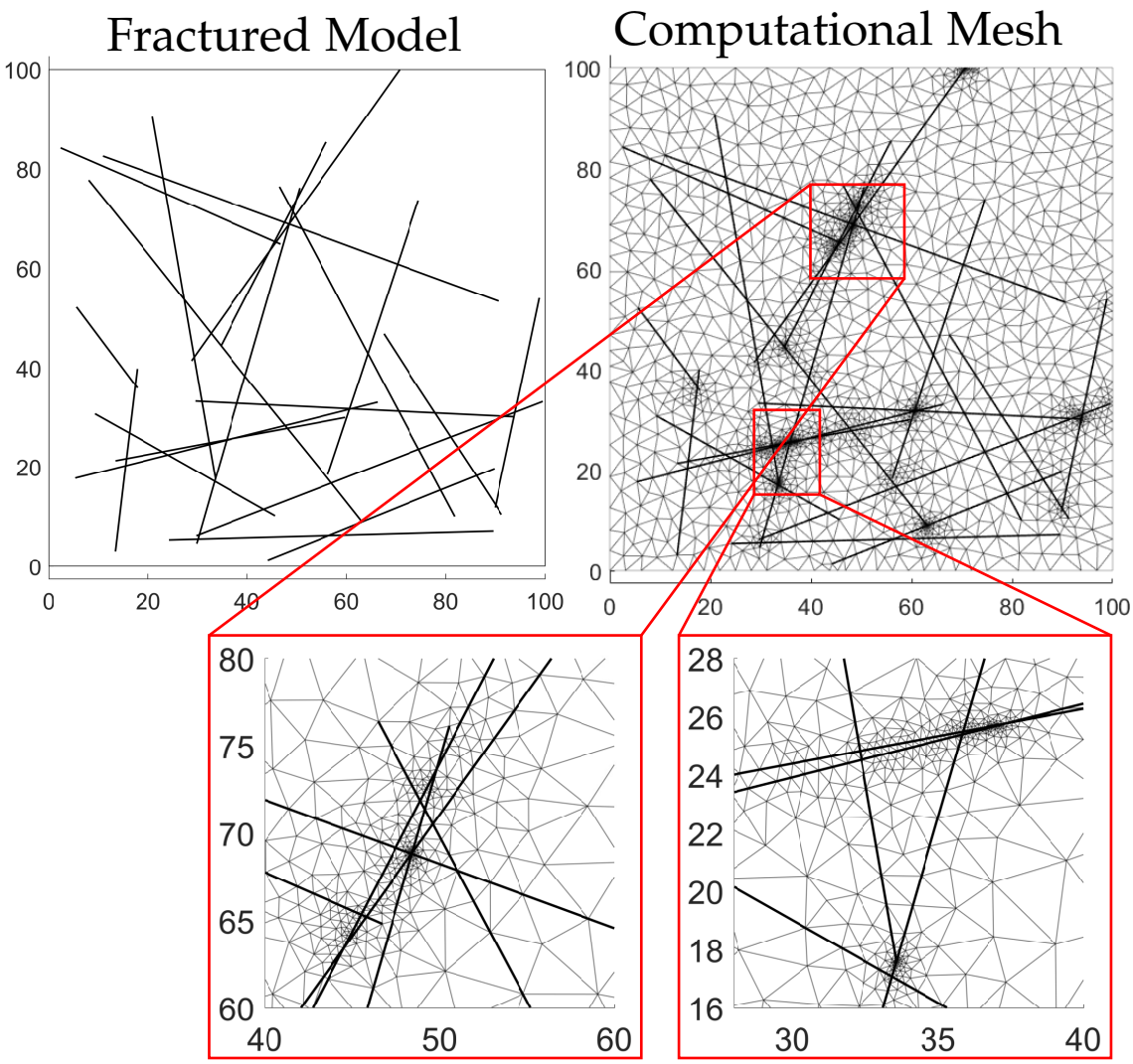

Figure 1.3: 2D Fractured model (left) and its corresponding triangular computational mesh. Conforming a mesh to a fracture system leads to very small mesh elements and elements with very high aspect ratios, which may cause numerical issues. Insets show areas of the mesh where this problem is particularly evident.

The EDFM method solves the meshing issue by embedding fractures within the simulation grid blocks (Lee et al., 2001) (see Figure 1.4). In the EDFM approach, the fractures do not need to match edges/faces of the computational grid and the meshing of the fractures is completely independent from the matrix. This allows, for instance, the utilization of cartesian or corner-point grids with arbitrary systems of fractures. In EDFM, conservation equations such as the ones defined by Equation 1.3 are solved for fracture and matrix separately, and matrix-fracture transfer is modelled as source terms in this equation. The transmissibility between fracture and matrix cells is calculated by considering the potential difference between the centroids of these cells and the cross-sectional area of the fracture cell within the grid block. The EDFM has been applied successfully to complex fracture patterns (see Figure 1.5). A currently known limitation of this approach is its inability to simulate rocks with anisotropic matrix per-

meability. Moreover, the pre-processing step that computes fracture-matrix and fracture-fracture intersections is computationally intensive.

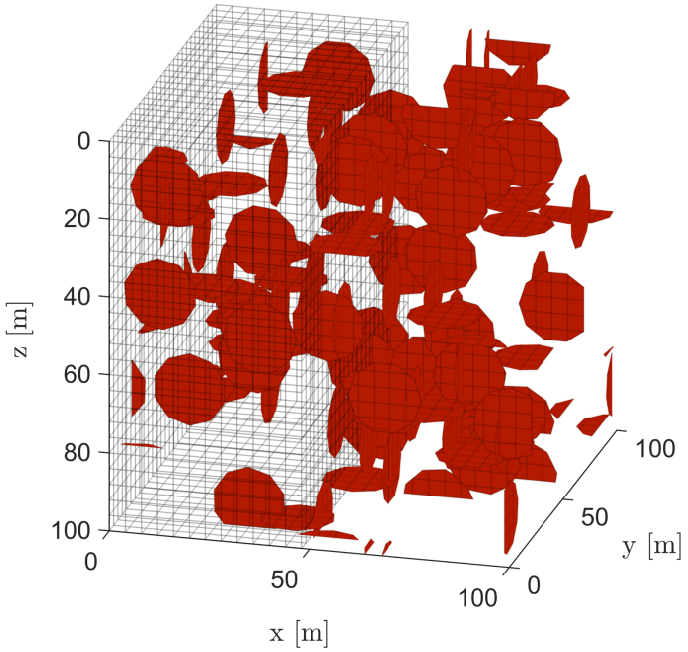

Figure 1.4: Network of fractures (ellipses in red) and simulation grid blocks in an EDFM model. Simulation grid blocks are shown for $x \leq 50$ m. Fracture and matrix models are courtesy of Daniel Wong, PhD Student at the Carbonate Reservoirs Group at Heriot-Watt University.

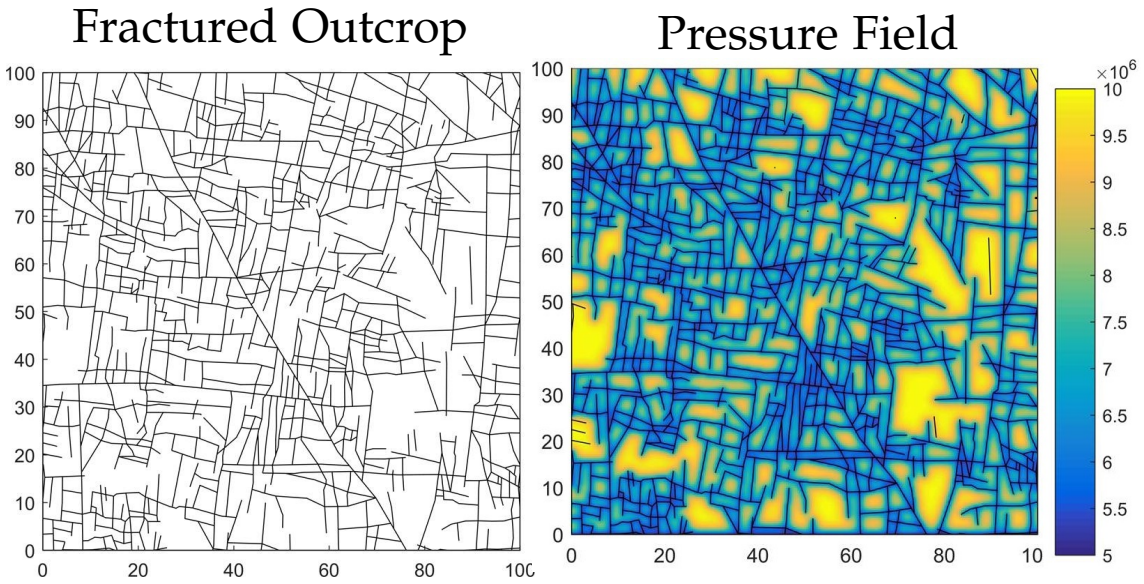

Figure 1.5: Single-phase flow simulation in a fractured outcrop with complex fracture system carried out in MRST. Map view of a fractured outcrop from Brazil (left) and pressure field (right). Courtesy of Daniel Wong, PhD Student at the Carbonate Reservoirs Group at Heriot-Watt University.

Multi-continuum methods address the main limitation of methods that represent fractures explicitly: 1) they do not generally allow the representation of all fractures at the reservoir scale due to computational complexities and 2) the

exact location of the fractures in the subsurface is generally not known. Multiple-continuum methods are popular in the reservoir simulation community due to their computational efficiency. They are based on the upscaling of the fracture system properties instead of representing each individual fracture. A key element of these upscaled methods are mass transfer functions that model the fluid exchange between fractures and matrix. The dual-porosity formulation and the impact of transfer functions in storage are discussed in details in the next section.

1.3.1 *Dual-porosity formulation*

The direct utilization of Equations (1.3) or (1.7) to simulate the flow in NFR's is prohibitively expensive for most realistic geological models. As discussed in the previous section, the utilization would require the explicit representation of all fractures by either conforming the simulation grid to them or by embedding them in the computational mesh. Dual-continuum approaches, on the other hand, do not require explicit fracture representation, and therefore, have been widely used in the past decades. In the dual-continuum approach, the fracture network is represented as a second continuum that is superposed to the rock matrix. The two continua are related by an upscaled transfer term T that models the mass transfer between the two continua. Hence, Equations (1.3) are extended by a set of conservation equations for the rock matrix:

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\phi_f \rho_{wf} S_{wf}\right) + \nabla \cdot \rho_{wf} \mathbf{q}_{wf} = \tilde{q}_w - T_w, \\
&\frac{\partial}{\partial t}\left(\phi_f \rho_{nf} S_{nf}\right) + \nabla \cdot \rho_{nf} \mathbf{q}_{nf} = \tilde{q}_n - T_n, \\
&\frac{\partial}{\partial t}\left(\phi_m \rho_{wm} S_{wm}\right) = T_w, \\
&\frac{\partial}{\partial t}\left(\phi_m \rho_{nm} S_{nm}\right) = T_n,
\end{aligned} \tag{1.8}$$

where the subscripts $\{f, m\}$ identify variables of the fracture and matrix continuum. The transfer rate term T_α models the rate of mass exchange of phase α between fracture and matrix per unit bulk volume. Several formulations for T_α have been suggested in the past decades. The first model was introduced in Warren and Root (1963) for a single compressible phase. The first multi-phase extension of this model was presented in Kazemi et al. (1976) and consists of scaling the fracture-matrix pressure potential by a transmissibility between the continua:

$$T_\alpha = \sigma \rho_\alpha k_m \lambda_\alpha \left(p_{\alpha f} - p_{\alpha m}\right), \tag{1.9}$$

where σ is the *shape factor*, which has units of $[1/L^2]$ and depends only on the matrix-block geometry. The shape factor encompasses the matrix-block area open for fluid exchange and the distance between the point in the matrix block where the matrix pressure is represented. There are several formulations for the shape factor; some of them are presented in Table 1.1. A limitation of this transfer function is that it does not capture gravity-induced transfer. A modification of this transfer function was suggested in Gilman (1986) and later in Sonier et al. (1988) to capture gravity-drainage effects. The transfer function proposed in Sonier et al. (1988) takes the following form:

$$
\begin{aligned}
T_w &= \sigma \rho_w k_m \lambda_w \left(\left(p_{wf} - p_{wm} \right) - \Delta\rho \left(h^e_{nf} - h^e_{nm} \right) g \frac{c}{2} \right), \\
T_n &= \sigma \rho_n k_m \lambda_n \left(\left(p_{nf} - p_{nm} \right) + \Delta\rho \left(h^e_{nf} - h^e_{nm} \right) g \frac{c}{2} \right),
\end{aligned}
\tag{1.10}
$$

with

$$
\begin{aligned}
h^e_{nf} &= \frac{S_{nf} - S_{nr}}{1 - S_{wr} - S_{nr}}, \\
h^e_{nm} &= \frac{S_{nm} - S_{nr}}{1 - S_{wr} - S_{nr}},
\end{aligned}
\tag{1.11}
$$

where c is the matrix-block height. With this correction, the gravity potential is split between wetting and non-wetting phases. This transfer function is implemented in the widely used commercial reservoir simulators Eclipse (Schlumberger, 2014) and IMEX (Computer Modelling Group, 2016). Notice that this family of transfer functions assumes that the matrix-fracture transfer is the same across all the matrix-block faces. In order to obtain a more accurate representation of the transfer physics, it was suggested in Quandalle and Sabathier (1989) that the transfer across each face is treated independently, and the total transfer is given by the sum over all matrix-block faces:

$$
T_\alpha = \frac{1}{abc} \sum_i T^i_\alpha,
\tag{1.12}
$$

where a, b and c are the block dimensions and T^i_α stands for the transfer of phase α across face i. The face transfer is written as

$$
T^i_\alpha = -\rho_\alpha k^i_m \lambda^i_\alpha \left(\Phi^i_{\alpha f} - \Phi^s_{\alpha m} \right),
\tag{1.13}
$$

where $\Phi^i_{\alpha f}$ stands for the potential of phase α at the fracture side of face i and $\Phi^s_{\alpha m}$ stands for the potential of phase α at the centre of the matrix block. The separation of the transfer across each face allows the treatment of block anisotropy and the representation of gravity drainage. In Abushaikha and Gosselin (2008) it was shown that while this transfer function represents fracture-matrix transfer

more accurately than previous formulations, it still fails to represent the transfer for all the physical mechanisms and fluid/rock properties. In Lemonnier and Bourbiaux (2010), the authors acknowledge that the computation of the transfer across each face of the matrix block does not fully solve the transfer-function modelling problem. They recommend that simulations of fully resolved matrix blocks are carried out in order to calibrate the transfer model.

Reference	Shape Factor
Kazemi et al. (1976)	$4\left(\frac{1}{a^2}+\frac{1}{b^2}+\frac{1}{c^2}\right)$
Kazemi et al. (1992); Lim and Aziz (1995)	$\pi^2\left(\frac{1}{a^2}+\frac{1}{b^2}+\frac{1}{c^2}\right)$
Lim and Aziz (1995)	$25.67/a^2$, with $a=b=c$
Coats (1989)	$24.0/a^2$, with $a=b=c$
Thomas et al. (1983)	$25.0/a^2$, with $a=b=c$
Ma et al. (1999)	$\sqrt{\frac{V_b}{\sum_{i=1}^{n} A_i/l_i}}$

Table 1.1: Summary of shape factors available in the literature. For the Ma et al. (1999) shape factor, V_b is the volume of the matrix block, A_i is the area of face normal to the i direction and l_i is the length of matrix-block in the i direction.

A fundamentally different approach is based in the "divide and conquer" (Lu et al., 2008) concept: each transfer mechanism (see Figure 1.2) is modelled separately and the total transfer between the continua is obtained by the sum of the transfers of the individual mechanisms. The exponential model of Aronofsky et al. (1958) has been regularly used to model the transfer due to spontaneous imbibition and gravity drainage separately (Kazemi et al., 1992; Schmid et al., 2011; Di Donato et al., 2006). It leads to a transfer function that is linear with respect to the saturation in the matrix:

$$T_\alpha = \beta\phi_m \left(S^*_{\alpha m} - S_{\alpha m}\right), \tag{1.14}$$

where β is the transfer rate coefficient that models the speed of the transfer process, and $S^*_{\alpha m}$ stands for the maximum saturation of phase α that the matrix block can hold. For gravity drainage $S^*_{\alpha m}$ is determined by the capillary-gravity equilibrium and for spontaneous imbibition $S^*_{\alpha m} = 1 - S_{nr}$. There are several expressions to obtain β for imbibition and drainage processes; unfortunately most of them with limitations. For spontaneous imbibition, the most accurate expression suggested in Schmid et al. (2011) is based on the analytical solution for spontaneous imbibition in a matrix block and captures reasonably well the general trend of experimental data available in the literature. However, it underestimates the early-time imbibition behaviour, that scales with the square root of time (see Figure 1.6). The limitations of this model are explored in more detail in Chapter 4. For gravity drainage, the formulation suggested in Di Donato et al.

(2006) is focused on oil and gas systems and relies on fitting parameters based on high resolution numerical simulations. In Chapter 3 we investigate this transfer function in greater detail. Finally, homogenization theory has been used as a formal framework to develop transfer functions that are more accurate than the standard ones based on fluid potentials (Arbogast, 1992; Tecklenburg et al., 2013).

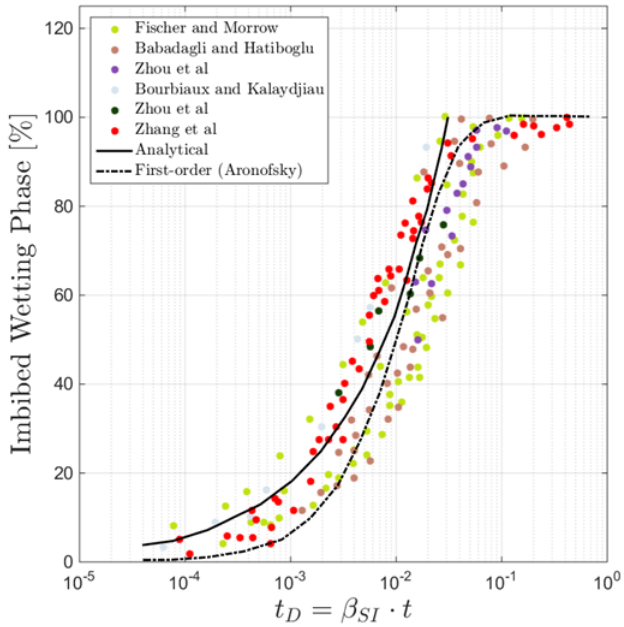

Figure 1.6: Data of several spontaneous imbibition experiments found in the literature (circles) with the exponential model of Schmid et al. (2011) and the analytical solution for early-time imbibition (McWhorter and Sunada, 1990; Schmid et al., 2011).

In summary, there are numerous models for modelling the matrix-fracture transfer in dual-porosity simulators. The presented list is not exhaustive and we refer to Lemonnier and Bourbiaux (2010), Abushaikha and Gosselin (2008) and Ramirez et al. (2009) for comprehensive reviews. To the best of our knowledge, most of the transfer functions either rely on the calibration against simulations of fully-resolved blocks or are based on simple pressure potentials and empirical shape factors that fail to capture the transfer physics by many orders of magnitude. This may pose a problem in the evaluation of CO_2 storage in NFRs, since transfer functions may have a very significant impact on the prediction of the location of the CO_2 plume. To illustrate this, we present the results of dual-porosity simulations of CO_2 storage in a geological model using different transfer models (Figure 1.7). The "physically-based transfer function" corresponds to the drainage transfer model developed in Chapter 3, and the "conventional transfer function" corresponds to the widely used model defined in Equations 1.10. The simulations were carried out using the Matlab Reservoir Simulation Toolbox

(MRST) (Lie et al., 2011), described in more details in Chapter 2. Data used in the simulations can be found in Appendix A. It is possible to see that the utilization of different transfer models for matrix drainage by CO_2 lead to significantly different amounts of CO_2 in the matrix and plume shapes in the fractures. Quantitatively, the difference in CO_2 mass between both simulation models amounts to approximately 0.375 $MtCO_2/year$, which is only two orders of magnitude smaller than the annual emissions of some small countries (according to information available in http://www.carbonmap.org/). Therefore, the utilization of a transfer function that correctly models the matrix-fracture fluid transfer can be decisive when determining whether a storage operation in NFRs is safe and economically viable.

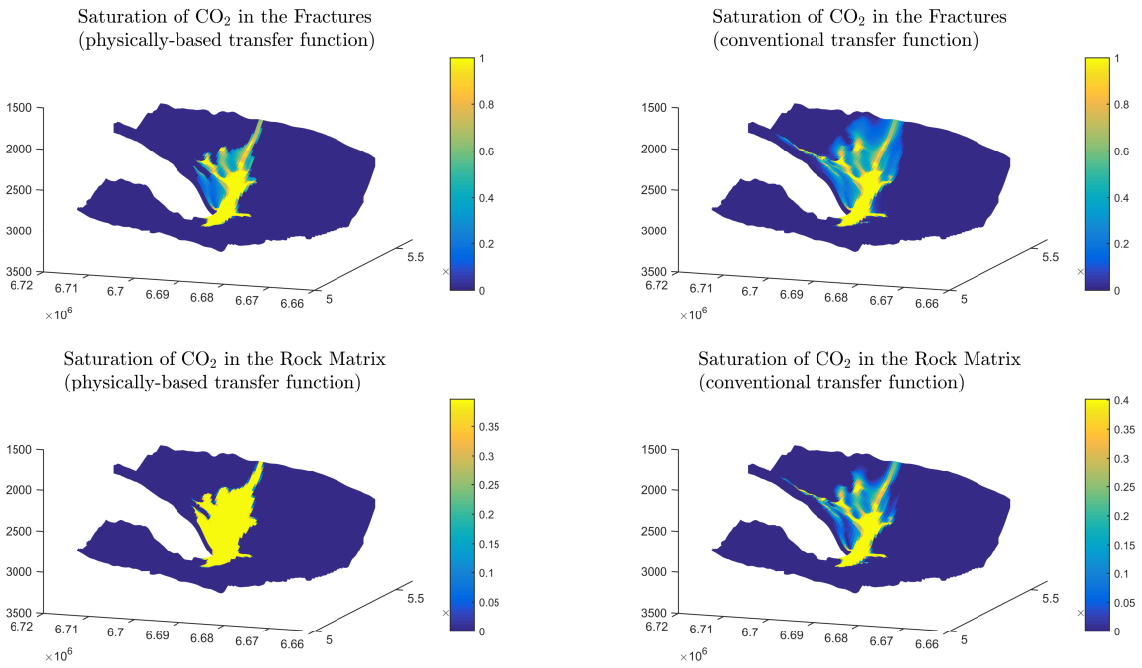

Figure 1.7: CO_2 plumes in the fracture and matrix after 120 years of injection using the physically-based transfer function (left) and the conventional transfer function (right). Simulation was carried out with MRST and data is available in Appendix A.

Current formulations of transfer functions, such as the ones discussed in this section, often fail to predict the drainage and imbibition dynamics accurately. The investigations carried out in this thesis show that the evaluation of CO_2 storage in NFRs with dual-porosity models requires new models to capture the relevant physical processes.

We consider the dual-porosity model defined by Equations (1.8) in this work to evaluate geological storage of CO_2 in NFRs. Dual-porosity models are applicable to naturally fractured reservoirs with a well-connected system of fractures and high permeability contrast between both media (Bourbiaux et al., 2002). We limit our analysis to such reservoirs. We do not perform simulations of flow in aquifer with explicitly represented fractures, since such simulations are computationally

expensive. This reinforces the value of dual-porosity approaches in obtaining understanding about the dynamics of multi-phase flow in fractured formations.

One of the key contributions of this thesis is the development of novel formulations for the transfer terms T_α that capture the multiphase transfer processes between fracture and matrix more accurately than existing formulations available in the literature.

1.4 OBJECTIVES AND STRUCTURE OF THIS THESIS

CO_2 capture and storage in subsurface formations is a promising technology to reduce the concentration of anthropogenic gases in the atmosphere that are causing the greenhouse effect. Naturally Fractured reservoirs (NFR's) are ubiquitous in sedimentary basins across the world, but have yet not been regarded as potential CO_2 storage sites. A series of concerns, such as the fast flow of CO_2 in the fractures, feed the scepticism towards CO_2 storage in fractured formations. To this point, no systematic study to evaluate storage in NFR's has been published. This thesis aims to fill this gap by providing tools and understanding to help evaluating the possibility of using NFRs to store CO_2. We achieve this objective by fulfilling the following specific goals:

- **Implementation of computational tools for simulation of two-phase flow in NFR's.** We provide open source computational tools that allow the simulation of CO_2-brine flow in fractured formations. These tools are: 1) the dual-porosity module in MRST and 2) the vertically integrated model with dual-porosity behaviour.

- **Improvement of dual-porosity models.** We identify the main weaknesses and improvement points of the currently existing formulations of transfer functions for dual-porosity models. We then develop novel transfer functions that target specifically the physics of CO_2-brine transfer during CO_2 storage operations in NFRs.

- **Simulation of CO_2 storage in conceptual models.** We look at conceptual representations of storage sites with realistic properties taken from the literature to understand under what conditions fractured reservoirs may be considered for storage.

This thesis is organized as follows:

- **Chapter 1. Introduction and Background.** We provide a brief introduction about CCS, simulation of flow in NFR's and the governing equations that model the multi-phase flow of CO_2 and brine in fractured and unfractured aquifers.

- **Chapter 2. Implementation of Multi-Continuum Models in the Matlab Reservoir Simulation Toolbox.** We describe the implementation of a dual-porosity module in MRST. We describe the object-oriented structure that allows us to create a framework for quick implementation and validation of transfer functions. We validate our implementation against block simulations carried out in the commercial reservoir simulator Eclipse.

- **Chapter 3. Modelling of CO_2-Induced Gravity Drainage.** We develop an estimate of the order of magnitude of the timescale of CO_2-induced matrix block drainage. This timescale consists of a model to estimate the time it takes to saturate a matrix block to equilibrium, and hence is an important component to understand the dynamics of CO_2 storage in NFR's. This model leads to a transfer function that accurately models gravity-drainage processes in the context of CO_2 storage. The application of this transfer function in the dual-porosity framework allows us to have a conceptual understanding of the impact of fractures during CO_2 storage operations. We then analyze how injection rates should be limited in order to avoid early spill of CO_2 (lost control of the plume) on a conceptual anticline model.

- **Chapter 4. Modelling of Capillary-Driven Spontaneous Imbibition.** We present a new hybrid approach to model counter-current spontaneous imbibition in 1D. This model combines analytically-based models for early- and late-time behaviour. The early-time behaviour is represented by a semi-analytical solution and the late-time behaviour uses an approximate exponential model with a universal transfer rate constant. A novel method that allows us to reconstruct the saturation profile inside the physical domain and to estimate the transition time is presented. The profiles of the imbibed volumes generated with this hybrid model are evaluated by comparing the results with a series of high-resolution numerical simulations with realistic physical properties arising from CO_2 storage applications. The imbibition model may be readily implemented in dual-porosity simulators where symmetry of the fracture geometry allows for a one-dimensional representation of the matrix blocks; for instance, if there is only one set of parallel fractures in the aquifer.

- **Chapter 5. CO_2 Storage Potential in Naturally Fractured Reservoirs.** We evaluate CO_2 storage potential in large aquifers in order to understand the key geological conditions that are favourable for storage. We develop a novel vertically-integrated model with dual-porosity behaviour that uses the gravity-drainage transfer function developed in Chapter 3. We perform a sensitivity analysis to understand the impact of basin temperature, depth

of the aquifer and its petrophysical properties in storage. We then consider realistic aquifers taken from the literature and evaluate the impact that a system of fractures would have in storage.

- **Chapter 6. Summary, Conclusions and Future Work** We present a summary of the main achievements of this thesis, general conclusions that can be drawn from the findings and suggestions for further developments in the field.

2

IMPLEMENTATION OF DUAL-POROSITY MODELS IN THE MATLAB RESERVOIR SIMULATION TOOLBOX

2.1 INTRODUCTION

In Chapter 1 we have introduced the dual-porosity models, which are typically used to simulate flow in NFR's due to their simplicity and efficiency. We have shown the results of simulations of CO_2 injection in a large aquifer with different transfer functions; both providing different results. Transfer functions are a key element in dual-porosity modelling. We will show in Chapter 3 and Chapter 4 that the accurate simulation of flow in NFRs with the dual-porosity models requires new models for matrix-fracture transfer. The implementation of these models is impossible in commercial reservoir simulators, such as Schlumberger's Eclipse (Schlumberger, 2014) and CMG's IMEX (Computer Modelling Group, 2016), as they do not provide access to the source code, nor an interface for quick prototyping of new transfer models. In contrast, the open-source Matlab Reservoir Simulation Toolbox (MRST) (Lie et al., 2011) allows for the implementation of new transfer models and comparison with existing formulations.

We develop a module in MRST, named `dual-porosity`, to provide an interface for rapid prototyping of dual-porosity new models. The `dual-porosity` module is distributed with MRST and all the developments described in this thesis are readily available for the interested reader. This chapter describes the structure of the `dual-porosity` module of MRST and presents validation examples that compare its results with commercial simulators and with single-porosity simulations.

2.2 MRST CODE STRUCTURE

MRST is an open-source platform to prototype and develop reservoir simulation algorithms and model concepts. Many different approaches of using MRST are possible ranging from using it as a standard reservoir simulator to a script-

ing environment with various useful routines that can be combined together as needed. It is built with a very slim core and additional features/routines are incorporated as modules. The toolbox is implemented in Matlab, but through the Matlab MEX functionality external routines (in C, C++ or Fortran) can be compiled and plugged into the code for increased performance.

MRST's basic grid structure is fully unstructured with nodes defining faces and faces defining cells, allowing full flexibility in the grid representation. Even cartesian and corner-point grids are treated as unstructured grids. Hence, an arbitrary mesh refinement is always possible as long as the discretisation concept allows for it. Most process-focused modules (such as the Black-Oil model) are thoroughly tested and implemented based on finite volume discretisation together with two-point flux approximation (TPFA) but other discretisations such as mimetic finite differences, virtual elements as well as multi-point flux approximation are available as well and can be adapted.

MRST is implemented as a flexible object-oriented environment, and with a high level of modularity that allows easy reusability of code. The main element of the object oriented approach is an *automatic differentiation* (AD) framework (Krogstad et al., 2015) that allows for fully implicit rapid prototyping and model-based and equation-based programming. Adding any equation within the AD structure is generally very simple.

Physical models are implemented in a *model* class. Any model class in MRST is inherited from a parent interface called `PhysicalModel`. In the object-oriented paradigm, an interface is a class with a set of methods that are empty, but that must be implemented by derived classes. Interfaces are useful for defining a specific behaviour of a code. In MRST's context, the code does not aim to solve a specific model, such as single-phase compressible flow, or three-phase black-oil; the implementation must be flexible to allow the implementation of any model. A class diagram that outlines the model concept in MRST is presented in Figure 2.1. The most relevant methods defined in the `PhysicalModel` interface are `getEquations` and `stepFunction`. The method `getEquations` corresponds to a function that returns the residual and the Jacobian of the equations given a state of field variables. This method is not implemented in this class, but any model subclass that extends `PhysicalModel` has to provide an implementation of this method. The method `stepFunction` advances a state of field variables by one step, by calling `getEquations` and solving the successive linear systems that define Newton's method.

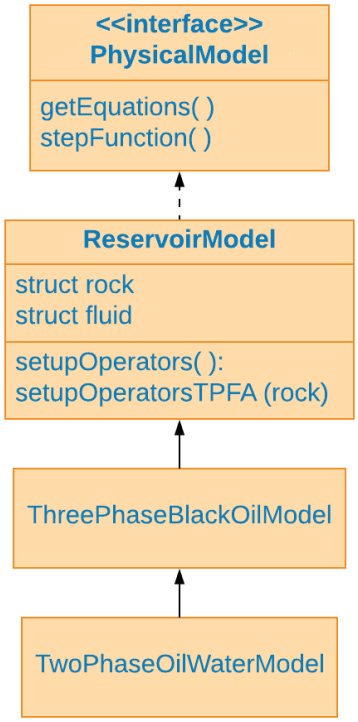

Figure 2.1: Simplified class diagram of the model structure of MRST.

The class `ReservoirModel` extends class `PhysicalModel` and provides functionality that is specific for models relative to reservoir simulation. This class has two important attributes: the `fluid` and the `rock` structures. These structures define fluid properties, such as viscosities and densities, and rock properties, such as porosity, permeability, relative permeabilities and capillary pressures. Saturation-dependent petrophysical functions are defined as function handles. This class also calls `setupOperatorsTPFA`, that calculates transmissibilities using the two-point flux approximation (Krogstad et al., 2015). Model-specific classes such as `ThreePhaseBlackOilModel` and `TwoPhaseOilWaterModel` implements the equations that are specific to these models.

The following code excerpt shows the implementation of the equations that solve two-phase immiscible flow (extracted from `equationsOilWater.m`).

```
% Conservation of mass for water
water = (s.pv/dt).*( pvMult.*bW.*sW - pvMult0.*bW0.*sW0 ) + s.Div(bWvW);

% Conservation of mass for oil
oil = (s.pv/dt).*( pvMult.*bO.*sO - pvMult0.*bO0.*sO0 ) + s.Div(bOvO);

eqs = {water, oil};
names = {'water', 'oil'};
types = {'cell', 'cell'};
```

For the dual-porosity module, these equations are extended by another set of conservation equations for the matrix, and source terms that model the transfer between the continua are added.

2.3 DUAL-POROSITY MODULE STRUCTURE

The extension of the MRST structure to incorporate dual-porosity models is implemented at the level of the `ReservoirModel`. Recall that this class stores two structures, `rock` and `fluid`, which hold rock and fluid properties. Moreover, this class is responsible for defining the list of primary variables (e.g. saturations, pressures) that may or may not be solved for in the derived classes. We have introduced a class named `DualPorosityReservoirModel` that defines extra structures for holding matrix properties, as well as extra variables to represent saturations and pressures of the matrix (see Figure 2.2). This class also has a variable that holds a generic instance of a transfer function, that represents the generic concept of a transfer model.

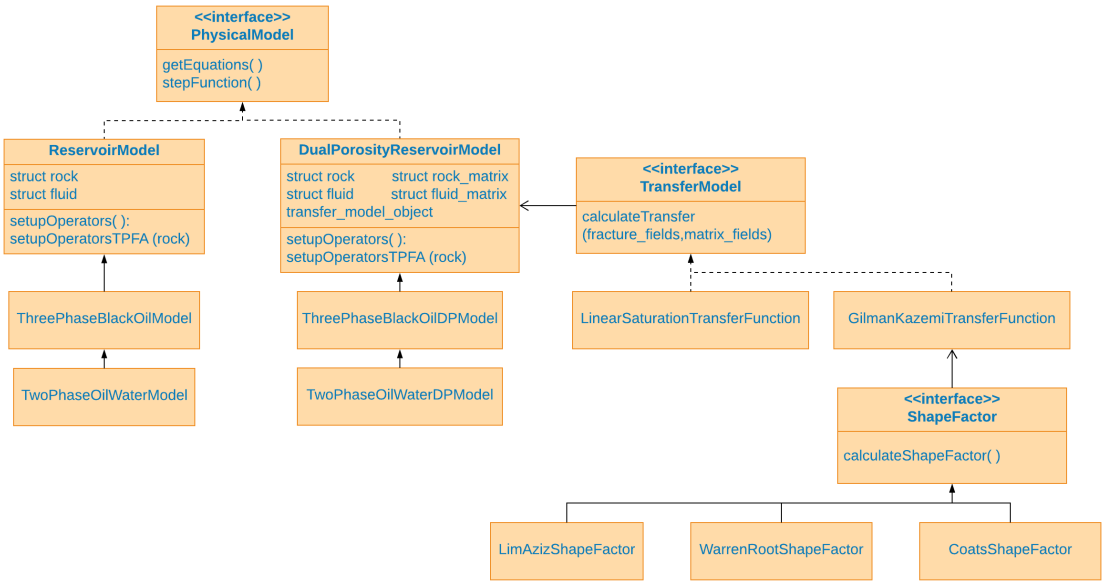

Figure 2.2: Simplified class diagram of the model structure of MRST with the dual-porosity module.

The `TransferModel` interface defines a dummy method that is called by all the derived classes: `calculateTransfer`. This method receives structures containing matrix and fracture fields and calculates the transfer for all phases. As seen in Chapter 1, some transfer models require shape factors (variable σ in Equation (1.9)). There are several models for the shape factor (see Table 1.1 for a summary of these models). Therefore, we have opted for a generic implementation, by defining a generic `ShapeFactor` interface that provides a method `calculateShapeFactor()` and can be specialized into specific models for σ.

The following code excerpt was extracted from `equationsOilWaterDP.m` and shows how the original two-phase model is extended to simulate dual-porosity behaviour.

```
%% Calculating transfer
vb = model.G.cells.volumes;

matrix_fields.pom = pom;
matrix_fields.swm = swm;
fracture_fields.pof = p;
fracture_fields.swf = sW;

transfer_model = model.transfer_model_object;

[Talpha] = transfer_model.calculate_transfer...
                  (model,fracture_fields,matrix_fields);

Twm = vb.*Talpha{1};
Tom = vb.*Talpha{2};
%%
%%
%% Extra code omitted here ...
%%
%%
% Conservation of mass for water - fracture
water_fracture = (s.pv/dt).*...
                                  ( pvMult.*bW.*sW - pvMult0.*bW0.*sW0 ) +...
                s.Div(bWvW);
water_fracture = water_fracture + Twm;

% Conservation of mass for oil - fracture
oil_fracture = (s.pv/dt).*...
                            ( pvMult.*bO.*sO - pvMult0.*bO0.*sO0 ) +...
               s.Div(bOvO);
oil_fracture = oil_fracture + Tom;

eqs{1} = water_fracture;
eqs{2} = oil_fracture;

% Add in any fluxes / source terms prescribed as boundary conditions.
eqs = addFluxesFromSourcesAndBC(model, eqs, ...
                                       {pW, p},...
                                       {rhoW,     rhoO},...
                                       {mobW,     mobO}, ...
```

```
                                        {bW, bO},  ...
                                        {sW, sO}, ...
                                        drivingForces);

% Matrix
% Conservation of mass for water - matrix
water_matrix = (s.pv_matrix/dt).*...
                            ( pvMultm.*bWm.*swm - pvMultm0.*bWm0.*swm0 );
water_matrix = water_matrix - Twm;

% Conservation of mass for oil - matrix
oil_matrix = (s.pv_matrix/dt).*...
                          ( pvMultm.*bOm.*som - pvMultm0.*bOm0.*som0 );
oil_matrix = oil_matrix - Tom;
```

Note that the calculation of the transfer is completely decoupled from the calculation of the residual of the equation. In other words, the model does not explicitly know how the transfer is calculated. This is a very desirable feature that allows the quick implementation of new transfer function and the utilization of these models within the AD framework without knowing specific details of the implementation of the simulation model. Several transfer functions (e.g. Kazemi et al. (1976) and Gilman (1986)) and shape factors (e.g. Coats (1989) and Lim and Aziz (1995)) are implemented in the dual-porosity module.

2.4 VALIDATION

We consider two validation cases in this chapter. The first evaluates if the dual-porosity module reproduces the results of a single-porosity system with explicit representation of fracture and matrix. The second compares the results of drainage of a single matrix block provided by Eclipse and by MRST using the same transfer function.

2.4.1 *Single-phase depletion*

We consider the depletion of a heterogeneous reservoir containing a single compressible phase. The reference model consists of a reservoir of size 500 m x 500 m x 10 m (see Figure 2.3). A constant and homogeneous porosity of $\phi = 0.2$ is considered for the entire model. A total of $N_x = 50$ blocks is used for discretization in the x direction. In the y direction, two layers of cells are considered. In the first layer, a well is placed in the leftmost cell. A low permeability ($k = 0.1$ mD) is assigned to the second layer, named here as "matrix". A significantly

higher permeability (k = 100 mD) is assigned to the layer with the well. This layer is named as "fracture" for simplicity; although it is essentially only a high permeability layer. The resident fluid has a viscosity of $\mu = 0.1\ Pa \cdot s$ and density $\rho = 1\ kg/m^3$. We assume a linear compressibility of $c_f = 10^{-7}\ Pa^{-1}$. The reservoir is at initial pressure of $p_i = 55.15\ MPa$ and the well has a constant bottom-hole pressure of $p_i = 6.89\ MPa$. Twenty time-steps of $\Delta t = 1$ day are used for the temporal discretisation. The expected behaviour is that the fracture will be quickly drained while the low-permeability matrix will be depleted much slower.

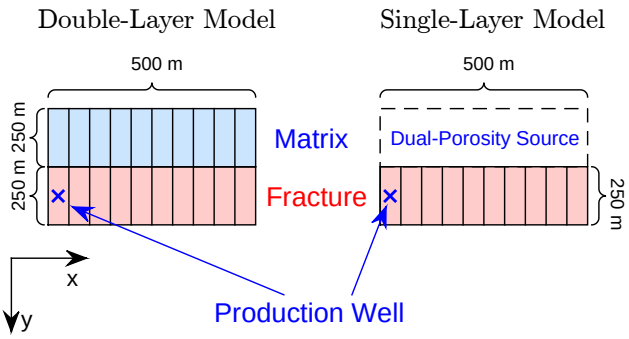

Figure 2.3: Geometry and boundary conditions for the single-phase depletion benchmark case.

We aim at matching the results of the two-layer system with a single-layer dual-porosity model. We use the Gilman-Kazemi transfer function (Equation 1.10). As shown in Chapter 1, the shape factor σ is defined as the the matrix-block area divided by the length between fracture and matrix centroids per unit volume of matrix block. In our synthetic model, grid cell dimensions match the matrix block dimensions, and σ is calculated as

$$\sigma = \frac{A_s}{L_{mf} V_b} = \frac{\Delta x \Delta z}{(\Delta y/2)\,(\Delta x \Delta y \Delta z)}, \tag{2.1}$$

where Δx, Δy and Δz are the dimensions of the grid cells in the x, y and z directions, respectively. Note that in a realistic porous medium, the Gilman-Kazemi transfer function would fail to capture the production rates coming from the matrix accurately. It is well known that such models are based on a pseudo steady-state assumption, and hence, would fail to capture the early-transfer regime (Zimmerman et al., 1993). However, since we are lumping the whole matrix block into just one simulation grid cell, this effect should not be visible in the two-layer model. Our objective with this benchmark is validation, and not verification. We aim at showing that a single-porosity system can be successfully modelled as a dual-porosity system with the proper choice of shape factors.

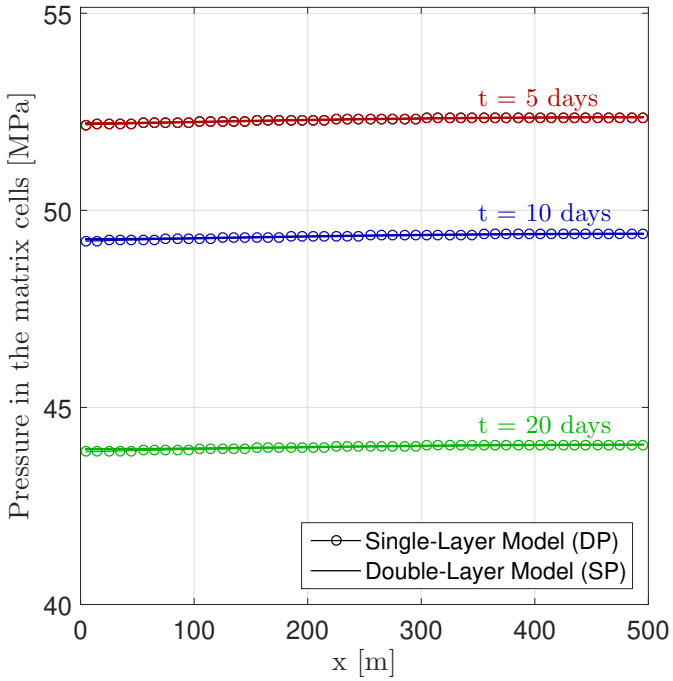

Figure 2.4: Pressure in the matrix in the single-layer and double-layer models at different times (t = 5 days, t = 10 days and t = 20 days).

Results from the comparative simulations are seen in Figure 2.4. The pressure in the matrix layer is monitored at different times during the reservoir depletion. A very good agreement is seen between the two-layer (solid lines) and the single-layer (round markers) solutions during the entire depletion process. The perfect match is not surprising: the Gilman-Kazemi transfer function is nothing more than a two-point flux approximation between fracture and matrix and our explicit representation of the two layers uses a two-point flux approximation as well. This simple benchmark problem validates the dual-porosity implementation for single-phase compressible flow and can even be used as an unit-test to ensure the consistency of the dual-porosity implementation during the development of the model.

2.4.2 *Gravity drainage of a single matrix block*

We now consider the validation of MRST against the commercial reservoir simulator Eclipse. Eclipse is widely used in industry and academia and is regarded as a reference software for simulation of flow in porous media. With this validation example, we aim at showing that the dual-porosity model matches the drainage of a single-matrix block when it is discretized by a single cell (i.e., a 0D system) when the Gilman-Kazemi transfer functions are used (note that Eclipse only provides two options, the Gilman-Kazemi and the Quandalle-Sabathier transfer function). This test problem is especially relevant in the context of this thesis, as

the developed drainage transfer functions (see Chapter 3) are compared to the Gilman-Kazemi transfer functions using MRST.

We consider a two-cell dual-porosity model. Two cells are necessary in this benchmark case because MRST does not handle a single cell model. The Eclipse configuration file is provided in Appendix B. The fracture cell is initially saturated with gas (non-wetting phase) and the matrix is initially saturated with oil (wetting phase). Both phases have very low compressibility. Oil has a density of $\rho_w = 1000\ kg/m^3$ and a viscosity of $\mu_w = 1$ cP. Gas has a density of $\rho_n = 600\ kg/m^3$ and a viscosity of $\mu_n = 0.04$ cP. Note that these properties are similar to brine and supercritical CO_2; when brine takes the role of oil. Each of the two grid cells has $\Delta x = 5$ m, $\Delta y = 5$ m and $\Delta z = 10$ m. We assume a single matrix block with the same dimensions of the grid cell. This leads to a shape factor of $\sigma = 4\left(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2\right) = 0.36\ m^{-2}$, using the model of Kazemi et al. (1976) (see Table 1.1). Linear relative permeabilities are assumed for both fracture and matrix and no residual saturations are considered here. We consider $p_c = 0$ in the fracture cells. For the matrix, we consider $p_c\left(S_w\right) = P_e S_w^{-0.5}$. We evaluate the drainage with respect to time for $P_e = 5$ kPa, $P_e = 10$ kPa and $P_e = 15$ kPa. The Eclipse data file in Appendix B corresponds to the case with $P_e = 5$ kPa.

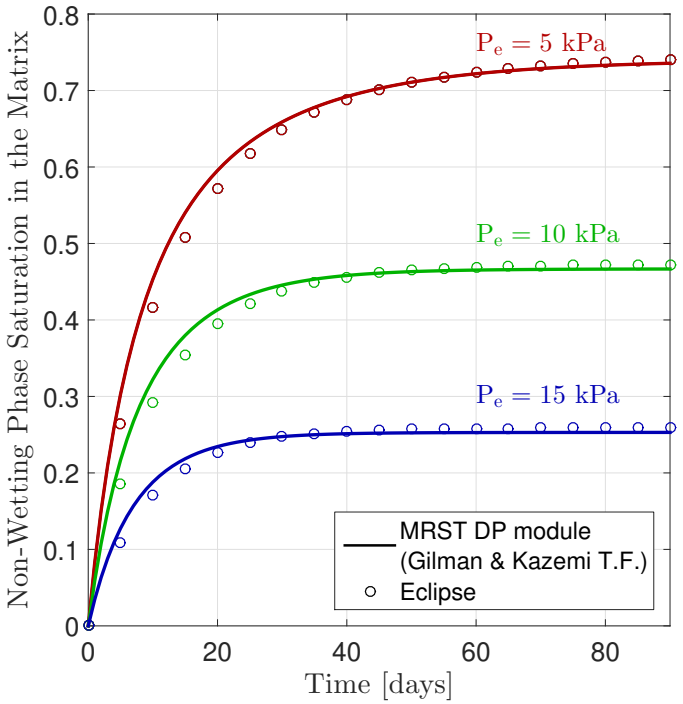

Figure 2.5: Saturation of non-wetting phase in the matrix cell with respect to time (MRST and Eclipse) for different capillary entry pressures ($P_e = 5$ kPa, $P_e = 10$ kPa and $P_e = 15$ kPa).

Figure 2.5 shows the non-wetting phase saturation in the matrix cells with respect to time. An excellent agreement is observed between the Eclipse and the MRST drainage curves. This figure shows that MRST is able to reproduce the drainage behaviour of the widely used Gilman-Kazemi transfer function on a

matrix block scale. Notice that the matrix blocks are not entirely drained by the gas. For instance, in the case with P_e = 15 kPa, only approximately 25% of the block is drained. This occurs because capillary forces oppose the drainage of the block and the final amount of gas that invades the block is determined by the capillary-gravity equilibrium. This concept is explained at detail in the next chapter, where a novel transfer function for gravity drainage is developed.

2.5 SUMMARY AND CONCLUSIONS

This chapter presented the implementation of a dual-porosity module in MRST. The module is published with the official release of MRST. A flexible and object-oriented implementation allows for easy adaptation and expansion of transfer models without knowledge of implementation details of the physical models.

The implementation was validated through two benchmarks: a single-phase depletion and block drainage. For the first benchmark, we have compared the results of a two-layer single porosity system with a single-layer dual-porosity system. For the drainage benchmark, we have compared MRST and Eclipse results of a single matrix block drainage with the standard Gilman-Kazemi transfer function. Both benchmarks confirmed the potential of the module to correctly simulate dual-porosity systems. The module presented in this chapter was used in the simulations carried out in this thesis.

3

MODELLING OF CO2-INDUCED GRAVITY DRAINAGE

3.1 INTRODUCTION

In this chapter, we develop a method to estimate the timescale of CO_2-induced matrix block drainage. The method consists of a model to estimate the time it takes to reach equilibrium between buoyant and capillary forces. The model also provides a transfer function that accurately models gravity-drainage processes in the context of CO_2 storage. The utilization of this transfer function in the dual-porosity framework allows us to develop a conceptual understanding of the impact of fractures on CO_2 storage.

The chapter is structured as follows. In Section 3.2, we present an analysis of storage capacity and drainage timescales for a single matrix block via numerical simulation of explicitly-represented and fully-resolved matrix blocks. In Section 3.3, we develop a model to estimate the timescale of matrix block drainage and a transfer function based on this timescale estimate. This model enables CCS simulations with dual-porosity models. In Section 3.4, this transfer function is evaluated in comparison with other transfer functions that are popular in commercial and open-source simulators. In Section 3.5, we consider an anticline geometry to evaluate the impact of fractures on field-scale storage using dual-porosity simulations. We also suggest to limit the injection rate to avoid the early spill of CO_2. Our calculations are based on the timescale estimate developed in Section 3.3. Finally, in Section 3.6, we provide a summary of the main results and contributions of this chapter, together with some general guidelines regarding CO_2 storage operations in fractured anticlines.

3.2 CAPACITY AND DRAINAGE TIMESCALES FOR A SINGLE MATRIX BLOCK

In Section 1.1 we discussed that the fast flow of CO_2 in the fractures creates isolated systems of brine-saturated blocks. These blocks are drained by CO_2 if buoyancy is large enough to overcome the capillary entry pressure. There are two important aspects to be analyzed in this transfer process. The first is the

maximum amount of CO_2 that can be stored in a matrix block that is surrounded by fractures saturated by CO_2. The maximum CO_2 volume gives a measure of the negative impact of fractures on CO_2 storage. The second is the time it takes for a matrix block that is being drained to reach equilibrium with the CO_2-saturated fractures. The drainage time provides a timescale for the CO_2 transfer to the matrix, and is important since a slow transfer process means that the CO_2 plume spreads farther in the fracture system and the potential for leakage is increased. In this section we analyze the results of fully-resolved numerical simulations of CO_2-brine drainage on three-dimensional matrix blocks.

3.2.1 *Setup for a single matrix block*

In order to understand the dynamics of drainage at the scale of a matrix block, a series of simulations were run using MRST to evaluate the amount of CO_2 stored in a block and the drainage timescale on a diverse number of CO_2 storage environments. A summary of the rock and fluid properties considered is provided in Table 3.1. As the drainage dynamics depend on the dimensions and aspect ratio of the matrix blocks, we consider two representative block geometries: a cubic block ($L_x = L_y = L_z = 10$ m) and a tall block, with the vertical length much larger than the horizontal dimensions ($L_x = L_y = 1$ m and $L_z = 20$ m) (see Figure 3.1). The cubic block represents a matrix block in a reservoir that has three perpendicular sets of fractures with sufficient aperture to prevent the formation of capillary bridges. Experiments show (Labastie, 1990) that for a surface tension of 24 mN/m capillary bridges do not form if apertures are greater than 0.3 mm. The cubic block has all its faces open for flow, representing the typical “sugar cube" geometry (Warren and Root, 1963). The tall block has only the lateral faces open for flow (no-flow boundary conditions at top and bottom faces), representing a reservoir with no horizontal fractures, or with strong capillary continuity across the vertical blocks. In this setting the no-flow boundary conditions at top and bottom faces represent the impermeable overburden and underburden of the formation and the block is long since it spans the entire thickness of the aquifer. This geometry if often referred to as “match stick" geometry in the literature (Ramirez et al., 2009). Blocks with similar aspect ratio are seen in fractured outcrops (Egya et al., 2018) and in the In Salah CO_2 storage project (Eiken et al., 2011). Note that we do not consider cases where the horizontal fracture spacing is much larger than the reservoir thickness. Fractures are modelled as boundary conditions for the matrix blocks. Fractures usually have very high permeability, and therefore injected CO_2 should quickly segregate in the fracture system and reach hydrostatic equilibrium (Guo et al., 2014). Hence, in both models, the open faces are prescribed with hydrostatic CO_2 pressure. The model was discretized

with uniform grid cells of $\Delta x = \Delta y = \Delta z = 10$ cm. A constant time step of $\Delta t = 0.1$ day was considered for all simulations. This discretization was shown to be sufficient to capture the dynamics for all the simulation cases.

It is important to reinforce that both sugar cube and match stick geometries have the lateral faces open for flow and therefore CO_2 may enter the matrix blocks through these faces. Typically, however, the higher pressure difference between fracture and matrix at the top of the block leads to a quick drainage of the top and to a subsequent displacement of the brine in a quasi-one-dimensional downwards flow. In fact, the three-dimensional numerical experiments presented in the next section show that the horizontal fluxes of CO_2 inside the matrix are often negligible. This observation is explored in the next sections to develop an approximate model for the drainage timescale based on the one-dimensional fractional flow equation.

	Case	Values
Environment	shallow-cold	ρ_w = 1012 kg/m^3, ρ_n = 714 kg/m^3, μ_w = 0.8 cP, μ_n = 0.05 cP
	shallow-warm	ρ_w = 998 kg/m^3, ρ_n = 714 kg/m^3, μ_w = 0.8 cP, μ_n = 0.05 cP
	deep-cold	ρ_w = 995 kg/m^3, ρ_n = 733 kg/m^3, μ_w = 0.38 cP, μ_n = 0.06 cP
	deep-warm	ρ_w = 945 kg/m^3, ρ_n = 479 kg/m^3, μ_w = 0.2 cP, μ_n = 0.04 cP
Sample	cardium#2 (sandstone)	k = 21 mD, S_{wr} = 0.42, n_w = 1.2, n_n = 1.3, k_{rn}^{max} = 0.13, ϕ_m = 0.16
	viking#2 (sandstone)	k = 21 mD, S_{wr} = 0.42, n_w = 1.7, n_n = 2.8, k_{rn}^{max} = 0.26, ϕ_m = 0.19
	wabamun#2 (sandstone)	k = 67 mD, S_{wr} = 0.57, n_w = 1.4, n_n = 2.1, k_{rn}^{max} = 0.19, ϕ_m = 0.15
	nisku#1 (carbonate)	k = 46 mD, S_{wr} = 0.33, n_w = 2.8, n_n = 1.1, k_{rn}^{max} = 0.18, ϕ_m = 0.1
	cooking-lake (carbonate)	k = 65 mD, S_{wr} = 0.48, n_w = 1.4, n_n = 5.6, k_{rn}^{max} = 0.07, ϕ_m = 0.1
Capillary Pressure Level	high-P_c	P_e = 40 kPa (match stick) and P_e = 25 kPa (sugar cube)
	low-P_c	P_e = 10 kPa (match stick) and P_e = 5 kPa (sugar cube)

Table 3.1: Physical parameters used to evaluate gravity drainage on a block scale. Each combination of environment, sample and capillary pressure level is considered, leading to a total of 40 simulation cases. Environments follow the definition of Nordbotten and Celia (2012). Samples were extracted from experiments (Bennion and Bachu, 2006). Two capillary pressure cases were considered respecting the maximum capillary entry pressure for CO_2 invasion, given the block size.

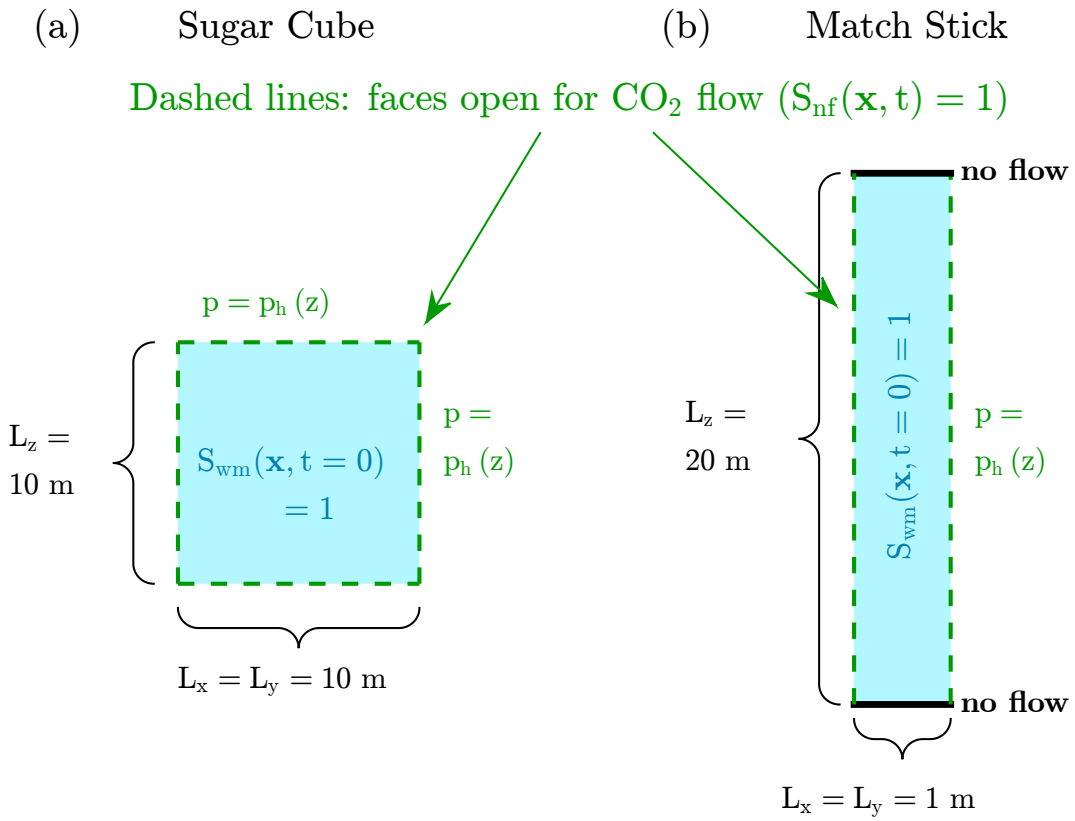

Figure 3.1: Conceptual picture of the block geometries and boundary conditions for (a) the sugar cube geometry and (b) the match stick geometry. Fractures are modeled as boundary conditions on the matrix blocks. Green dashed lines represent open boundary faces, open for CO_2 flow, while solid black lines in the match stick model represent no-flow boundaries. Only a projection in the x-z plane is shown, but geometries are 3D with $L_y = L_x$.

To cover representative scenarios for CO_2 storage, we consider four different storage environments, representing formations at shallow and deep depths in warm or cold basins (Nordbotten and Celia, 2012). These cases are named shallow-cold, deep-cold, shallow-warm and deep-warm. “Shallow" formations are located at 1-km depth, while “deep" formations at 3-km depth. “Cold" basins have a surface temperature of 10 ^{o}C and geothermal gradient of 25 ^{o}C/km, while “warm" basins have a surface temperature of 20^{o}C and geothermal gradient of 45^{o}C/km. We consider low-salinity brines. The corresponding densities and viscosities of the fluids for each of the four environments are taken from Nordbotten and Celia (2012) and given in Table 3.1. Representative values for porosity, residual brine saturation, absolute and relative permeability parameters are taken from five samples (two sandstones and three carbonates) from Bennion and Bachu (2006). We keep their sample names and refer to them as viking#2, cardium#2, wabamun#2, nisku#1 and cooking-lake. The parameters are given in Table 3.1. Since there is no capillary pressure data available for the five selected samples, we consider two capillary pressure levels, a “low-P_c" and a “high-P_c" case (see Table 3.1). Capillary pressure depends on a series of parameters that

are generally uncertain and difficult to measure in-situ, such as surface tension and contact angle between the phases. Therefore, we have decided to cover these two possible scenarios. In the remainder of this text, each simulation case will be labelled as {sample}-{environment}-{pc level} (e.g. cardium#2-deep-cold-low-P_c). A total of 2 shapes $\times$ 4 environments $\times$ 5 samples $\times$ 2 P_c levels = 80 simulations were considered in this analysis.

3.2.2 *Capacity and drainage timescales from numerical simulations*

In this section, we discuss the results of the numerical simulations by analyzing the maximum volume of CO_2 that invaded the block when the fluids reach an equilibrium, denoted as $V_{CO_2}^{max}$, the maximum mass of CO_2 in the block at this point, denoted as $M_{CO_2}^{max}$, and the time it takes to reach equilibrium, denoted as t_{95}. We take t_{95} from the numerical simulations as the time when the volume of CO_2 inside the block corresponds to 95% of $V_{CO_2}^{max}$. As the lateral faces are open for flow and at hydrostatic pressure of CO_2, in equilibrium the phase pressures are hydrostatic and their difference is compensated by capillary pressure. The final saturation profile is calculated analytically by evaluating the inverse of the capillary pressure at the difference between hydrostatic pressures for each depth: $S_n^{\infty}(z) = 1 - p_c^{-1}\left(\Delta\rho g(L_z - z)\right)$, where $\Delta\rho = \rho_w - \rho_n$ and the z axis denotes depth. This gives a maximum volume of $V_{CO_2}^{max} = \phi_m L_x L_y \int_{z=0}^{z=L_z} S_n^{\infty}(z)dz$. The maximum volume is calculated analytically based on the rock and fluid properties for each simulation case. The maximum mass is calculated by multiplying the maximum volume by the density of CO_2, $M_{CO_2}^{max} = \rho_{CO_2} V_{CO_2}^{max}$. We also define the pore volume that is effectively used for storage, $V_{CO_2}^{\%} = V_{CO_2}^{max}/(\phi_m(1 - S_{wr})V_b)$, where V_b is the volume of the block. This parameter allows us to evaluate the storage loss due to the presence of fractures.

Figure 3.2 shows the results of the numerical simulations for the sugar cube block. We restrict the analysis in this section to this block shape, noting that it also applies for the match stick geometry. The circles are inversely scaled according to the capillary pressure level (small circles correspond to high-P_c and large circles correspond to low-P_c). The circle fill colors correspond to different environments (shown in the legend of the picture), while the color of the edges correspond to different samples (shown in the picture).

The drainage time, t_{95}, shows a large variation across the different cases, ranging from $t \approx 10$ days to $t \approx 355$ days. It is therefore an important metric to be observed during storage operations. In the low-P_c cases, the largest drainage times are seen in the viking#2 and nisku#1 samples, due to their larger effective pore volume and lower permeability. The shallow-cold environment leads to higher times compared to the deep-cold, which is more evident for the nisku#1

sample ($t \approx 355$ days vs. $t \approx 150$ days). This is due to the larger viscosity difference between CO_2 and brine in deep-cold compared to shallow-cold environments ($\mu_n/\mu_w \approx 0.157$ vs. $\mu_n/\mu_w \approx 0.06$). The deep-warm environment shows the largest value of viscosity ratio, which explains why the yellow circles are concentrated on the left half of the plot ($\mu_n/\mu_w \approx 0.2$). The wabamum#2 sample has similar properties to that of the cooking-lake. Hence, they show similar maximum masses and drainage times for each environment in the low-P_c cases. However, a high contrast in t_{95} is seen between these samples for the deep-warm, high-P_c cases ($t_{95} \approx 11$ days vs. $t_{95} \approx 133$ days). The unfavourable conditions seen in the cooking-lake sample are due to very high Corey exponent for the CO_2 relative permeability ($n_n = 5.6$), that slows down drainage. This effect is less evident when buoyancy forces are very strong. In fact, in hydrocarbon recovery operations, the assumption of $\mu_w \gg \mu_n$ usually leads to an equation for the non-wetting phase invasion that completely disregards the influence of the non-wetting phase (Hagoort, 1980). That is, if buoyancy is high enough and the viscosity of the wetting phase is much larger than the viscosity of the non-wetting phase, the wetting phase dominates the flow and the non-wetting phase only occupies the void space without creating any resistance to flow. However, this is not always the case in CO_2 storage operations. In supercritical state, CO_2 has its viscosity increased, decreasing the viscosity contrast between CO_2 and brine. This is reflected in the high drainage times seen in the cooking-lake-high-P_c case due to high n_n.

While $M_{CO_2}^{max}$ changes for each sample due to different porosities and S_{wr}, $V_{CO_2}^{\%}$ changes only with the environment and the capillary pressure level, as this parameter is normalized by the effective pore volume for each case. One of the key parameters to understand the behavior of the block stored mass and volume is the density difference between brine and CO_2. It determines the maximum penetration depth of the CO_2 front. The shallow-warm environment has a higher $\Delta\rho$ ($\Delta\rho = 732\ kg/m^3$), while the deep-cold has a lower $\Delta\rho$ ($\Delta\rho = 222\ kg/m^3$). Therefore, the shallow-cold and deep-cold environments show different values for the CO_2 storage volume: $V_{CO_2}^{\%} \approx 85\%$ and $V_{CO_2}^{\%} \approx 60\%$, respectively, for the low-P_c case (Figure 3.2(b)). On the other hand, cold basins lead to the highest values of CO_2 density, as ρ_{CO_2} increases with decreasing temperature ($\rho_n = 714\ kg/m^3$ and $\rho_n = 733\ kg/m^3$ for shallow-cold and deep-cold environments). Hence, more CO_2 is stored. This explains the behaviour of the shallow-cold and deep-cold environments in the upper part of Figure 3.2(a). In short, high ρ_{CO_2} implies low $\Delta\rho$ and therefore low $V_{CO_2}^{\%}$, but also implies high $M_{CO_2}^{max}$. This is an interesting competing behavior of the CO_2 density that is only seen in fractured formations.

In summary, this section analyzed the behavior of three important parameters to be regarded in CO_2 storage operations in NFR's: $V_{CO_2}^{\%}$, $M_{CO_2}^{max}$ and t_{95}.

While the first two parameters are easily calculated using the rock and fluids properties, the drainage timescales were post-processed from the numerical simulations. Since running fully-resolved block simulations may be prohibitive for real formations, the development of an analytical estimate for t_{95}, the drainage timescale, is an important contribution of this work and is shown in the next sections.

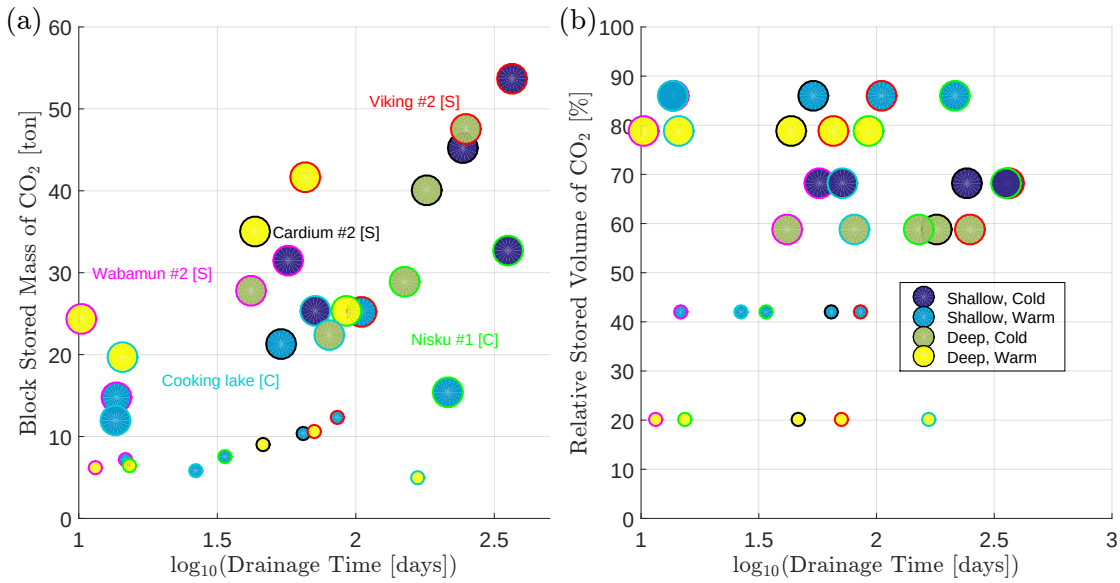

Figure 3.2: Analysis of the maximum CO_2 storage and drainage timescales on a single block for the sugar cube geometry. (a) Drainage time vs. stored mass of CO_2; (b) Drainage time vs. relative stored volume of CO_2. The relative volume provides a percentage measure of the pore volume not effectively used for storage due to the presence of a fracture system. Small circles correspond to high-P_c cases.

3.3 ESTIMATION OF DRAINAGE TIMESCALE FROM FRACTIONAL FLOW

In this section, we investigate the saturation equation for incompressible two-phase flow (Equation (1.7)) to develop a timescale estimate for the drainage process. We consider that drainage occurs essentially as a vertical downwards displacement, as observed in experiments described by Hagoort (1980) and Pedrera et al. (2002), and later used by Di Donato et al. (2006) and March et al. (2016) to develop transfer functions for gravity drainage. Therefore, we rewrite Equation (1.7) below assuming one-dimensional downwards displacement of the non-wetting phase and omitting the source terms:

$$\phi_m \frac{\partial S_w}{\partial t} = -\frac{\partial}{\partial z}\left(\overbrace{f_w}^{\Psi} q_t + \overbrace{k f_w \lambda_n \frac{dP_c}{dS_w}}^{\Pi} \frac{\partial S_w}{\partial z} + \overbrace{k f_w \lambda_n \Delta\rho g}^{\Gamma} \right). \tag{3.1}$$

We note that the groups Ψ, Π and Γ are functions of the saturation only apart from $q_t = q_t(t)$ and $\frac{\partial S_w}{\partial z}$, which are non-local and depend on z and t. We estimate $\frac{\partial S_w}{\partial z} \approx \frac{\Delta S_w}{H}$ and assume that we can find a suitable estimator for the total flux $q_t(t) \approx \bar{q}_t$. Given these approximations, the groups are a function of S_w only and the spatial differential is written as $\partial/\partial z = (\partial S_w/\partial z)(d/dS_w)$ via the chain rule. Hence, a timescale for the saturation change is estimated by linearizing each of the terms of Equation (3.1):

$$\begin{aligned}
&\phi_m \frac{\Delta S_w}{\tilde{t}} \approx \frac{\Delta S_w}{H} \frac{d}{dS_w}\left(\bar{q}_t \Psi + \frac{\Delta S_w}{H}\Pi + \Gamma\right) \Rightarrow \\
&\frac{1}{\tilde{t}} \approx \frac{1}{\phi_m}\frac{1}{H}\left(\bar{q}_t \frac{d\Psi}{dS_w} + \frac{\Delta S_w}{H}\frac{d\Pi}{dS_w} + \frac{d\Gamma}{dS_w}\right),
\end{aligned} \tag{3.2}$$

where ΔS_w is estimated by $\Delta S_w \approx 1 - S_{wr}$. H is the maximum penetration depth of CO_2 in the matrix block, that is, the depth inside the matrix block where the difference between the phase pressures of brine and CO_2 equals the capillary entry pressure. It is calculated as $H = L_z - P_e/(\Delta\rho g)$. This parameter is a natural length scale for the drainage process, as most of the drainage happens until the CO_2 front reaches this depth. This approach follows the developments of Di Donato et al. (2006), but with a key difference regarding the assumptions about the fluid properties: Di Donato et al. assume that $\mu_w \gg \mu_n$, which reduces the equation to a form that does not include the group Ψ and where the groups Π and Γ do not include the mobility of the non-wetting phase (λ_n); this assumption is not valid for CO_2-brine systems. We have shown in the last section that in some scenarios relevant to CO_2 storage applications the wetting phase properties do alter the behavior of the drainage process. Therefore, we do not make any assumptions regarding the fluid properties and aim to derive a general estimate of the timescale for any set of properties. The saturation derivatives in Equation (3.3) are estimated by considering the maximum of each group over the saturation range as a characteristic value. We therefore write:

$$\frac{d\Psi}{dS_w} \approx \frac{\max_{S_w}(\Psi)}{\Delta S_w}, \tag{3.3}$$

$$\frac{d\Pi}{dS_w} \approx \frac{\max_{S_w}(\Pi)}{\Delta S_w}, \tag{3.4}$$

$$\frac{d\Gamma}{dS_w} \approx \frac{\max_{S_w}(\Gamma)}{\Delta S_w}. \tag{3.5}$$

To close the model, a characteristic value for $\bar{q}_t$ is needed. We note that the total flux is constant in z but changes over time. It is highest at the onset of the drainage process and approaches zero when drainage is finished. We therefore use the maximum total flux to provide an estimate to q_t ($\bar{q}_t = q_t(0)$). Isolating q_t

in Equation (1.4), integrating in space and assuming hydrostatic pressure in the non-wetting phase at the boundaries, we obtain:

$$q_t\left(t\right) = \frac{1}{\int_0^{L_z} \frac{1}{k\lambda_t} dz} \left[-\rho_n g L_z - p_c(z=0) + \int_0^{L_z} f_w \rho_w g + f_n \rho_n g - \lambda_n \frac{\partial p_c}{\partial z} dz \right], \quad (3.6)$$

which gives us, assuming $S_w\left(z, t=0\right) = 1$:

$$\bar{q_t} = q_t\left(t=0\right) = \frac{k k_{rw}^{\max}}{L_z \mu_w} \left(\Delta\rho g L_z - p_c(z=0)\right). \quad (3.7)$$

The capillary pressure at the top of the domain can be estimated by considering a high value of brine saturation, say $S_w = 0.9$, noting that at the onset of the drainage process only a small amount of non-wetting phase will have invaded the domain. The total flux estimate closes the model, and its performance will be assessed by using it in a transfer function that is tested against the high-resolution simulations described in the previous section. The evaluation of the saturation-dependent functions at their maximum may lead to an over-estimation of the transfer rate for some cases, particularly in the late-time regime of the drainage process. However, as shown in Section 3.5, the transfer function that is developed in the next section using this timescale predicts with good accuracy the drainage of the blocks for a large part of the drainage time.

3.3.1 *Development of a new transfer function for CO_2 gravity-drainage*

Most commercial simulators (e.g. Eclipse and IMEX) and scientific publications (e.g. (Ahmed Elfeel et al., 2016; Bech et al., 1991; Beckner et al., 1991)) rely on the work of Gilman (1986) to model gravity-induced drainage transfer functions $\{T_w, T_n\}$ (Equation 1.8), sometimes applying a discretization of the matrix blocks (Pruess, 1985; Beckner et al., 1991). These transfer functions model the matrix-fracture flow as proportional to the potential difference between the two continua. Mathematically, this translates to:

$$\begin{aligned} T_n &= \sigma \frac{k_m k_{rn}}{\mu_n} \left(p_n^f - p_n^m + \Delta\rho (h_n^f - h_n^m) g \frac{L_z}{2} \right), \\ T_w &= \sigma \frac{k_m k_{rw}}{\mu_w} \left(p_w^f - p_w^m - \Delta\rho (h_n^f - h_n^m) g \frac{L_z}{2} \right), \end{aligned} \quad (3.8)$$

where σ is the shape factor, that accounts for the matrix area that is open for flow, k_m is the permeability of the matrix and $k_{r\alpha}$ are the relative permeabilities, that are typically evaluated upstream. However, despite intuitive and simple to implement in a dual-porosity simulator, it has been noted by other authors (Abushaikha and Gosselin, 2008) that this model fails to capture gravity drainage

processes accurately, specifically if the blocks are tall. Moreover, it is important to note that the transfer functions defined by Equation (3.8) do not converge to the correct limit $V_{CO_2}^{max}$ as the transfer ceases, and will therefore over-estimate the final storage of CO_2 at the matrix-block scale. This is typically corrected using *pseudoization* techniques that adjust the final amount of non-wetting phase that drains the matrix, at the cost of misrepresenting the drainage dynamics even more (Abushaikha and Gosselin, 2008; Schlumberger, 2014). As shown later in this section, the mismatch between the storage results given by high-resolution block simulations and the results given by Equation (3.8) are specially high when considering rock and fluid properties relevant to CO_2 storage applications.

The timescale $\tilde{t}$ defined in Equation (3.3) is used to develop an improved transfer function that captures the dynamics of CO_2-induced gravity-drainage. We start by defining the transfer rate coefficient β as:

$$\beta = \frac{1}{\tilde{t}}. \tag{3.9}$$

We then use β on an exponential model, as initially proposed by Aronofsky et al. (1958) and later on used by Kazemi et al. (1992), Schmid and Geiger (2012) and Schmid et al. (2013) to model spontaneous imbibition and by Zhou et al. (2014) to model diffusive transfer of CO_2 in fractured reservoirs. We therefore write the relative transfer of CO_2 to the matrix by gravity drainage as:

$$\frac{V_{CO_2}(t)}{V_{CO_2}^{max}} = 1 - e^{-t/\tilde{t}} \equiv 1 - e^{-\beta t}. \tag{3.10}$$

Differentiation of this expression with respect to time and multiplication by a smooth activation function $F\left(S_{nf}\right)$ leads to:

$$\phi_m \rho_{nm} \frac{\partial S_{nm}}{\partial t} = T_n = \phi_m \rho_{nm} \beta F\left(S_{nf}\right)\left(S_{nm}^{\max} - S_{nm}\right), \tag{3.11}$$

where S_{nm}^{max} is the average saturation corresponding to the maximum volume of CO_2 a block can store. The activation function ensures a smooth activation of the transfer only when CO_2 is available in the fracture cell. It is defined (Lu et al., 2008) as:

$$F\left(S_{nf}\right) = \frac{1 - e^{-\sqrt{k_f^*/k_m} S_{nf}}}{1 - e^{-\sqrt{k_f^*/k_m}}}, \tag{3.12}$$

where k_f^* is a measure of the permeability in the fracture cell, taken here as $max\{k_{fx}, k_{fy}, k_{fz}\}$. The activation function ensures the transfer starts when the fractures are saturated by CO_2. The transfer ceases exactly when the block is saturated by the maximum amount of CO_2. It is important to emphasize that a

general assumption of transfer functions like the one defined above is that all the matrix blocks within a simulation grid block are similar and are surrounded by CO_2 once the plume arrives at this point. This is a common assumption done in the derivation of transfer functions for dual-porosity models, that is particularly reasonable when the vertical dimensions of the block are larger than the horizontal ones, such as in the In Salah CO_2 storage project (Eiken et al., 2011; Iding and Ringrose, 2010).

As phases are assumed incompressible, summation of the phase equations in the matrix (last two equations of 1.8) leads to $T_w = -T_n$. We note that this expression is very similar to the one suggested by Di Donato et al. (2006), but with a different transfer rate coefficient (parameter β). Di Donato et al. suggests the calculation of β as:

$$\beta = \left(\frac{1}{n_w}\right)\omega_g + (r-1)\,\omega_c, \tag{3.13}$$

where r is the gravity/capillary ratio, given by $r = \Delta\rho g L_z / P_e$, and ω_g and ω_c are characteristic timescales for gravity and capillary forces, given by $\omega_g = (k k_{rw}^{max} \Delta\rho g)/(\phi_m H)$ and $\omega_c = (k k_{rw}^{max} n_p P_e)/(\phi_m H^2)$. The $(1/n_w)$ and $(r-1)$ factors multiplying each timescale were derived empirically by matching the transfer expression to high-resolution one-dimensional block simulations. We note that the expression derived in this chapter contains no parameters that were fitted to numerical simulations, in contrast to other approaches.

3.4 Evaluation of the gravity-drainage transfer function for a single matrix block

Figures 3.3(a) and 3.4(a) show a comparative analysis between the transfer model proposed in this chapter and the one proposed by Gilman (1986). The curves corresponding to the latter were generated by running a 0D dual-porosity model of single fracture and matrix cells, and tracking the saturation of CO_2 in the matrix block over time. No correction of the final volume of CO_2 in the block was applied here; the transfer is implemented as shown in Equation(3.8). For the sugar cube geometry (Figure 3.3(a)), each case has the same position as in Figure 3.2, but the circles are now scaled by the relative integral error of the transfer model defined by Equation (3.10). The same applies for the match stick geometry. The relative integral error is defined as $E_{model} \equiv \int |I_{num} - I_{model}|\, dlogt / \int I_{num} dlogt$, where I_α is the normalized stored volume of CO_2 for the model α. Graphically, for each model, E_{model} represents the ratio between the area between the model and numerical curves and the area below the numerical curve in Figures 3.3(b) and 3.4(b). For $E_{model} = 0$, the match is perfect. The error of Gilman's expression

is represented as empty circles in the same positions of the solid circles, and also scaled by E_{model}. Actual error values are shown in the figure for the cases with high t_{95} in the plot for reference. For the color pattern used to identify each circle on the chart, please refer to Figure 3.2.

Figures 3.3(a) and 3.4(a) show that the model proposed in Equation (3.10) provides much more accurate results compared to the one given by Equation (3.8) for all cases (solid circles are always inside the empty circles). This is especially true for the match stick cases, which consist of fracture geometries extremely favourable to storage. The transfer model given by Equation (3.8) misrepresents the storage dynamics by a factor as high as $\approx$ 325% for the match stick geometry and $\approx$ 139% for the sugar cube geometry, both for the cardium#2-shallow-cold-low-P_c case. The maximum error incurred by the proposed model is $\approx$ 23% for the match stick geometry (viking#2-shallow-cold-high-P_c case) and $\approx$ 16% for the match stick geometry (cooking-lake-deep-warm-high-P_c case). The best results are achieved for the high-storage cases (top part of the chart in Figures 3.3(a) and 3.4(a)). Figures 3.3(b) and 3.4(b) show representative storage curves for some high storage cases. In these plots, the model of Di Donato et al. (2006) is also shown. The proposed model (dashed lines) seems to represent, with good accuracy, the whole extent of the drainage dynamics for most of the cases. The nisku#1 cases have a higher value of the Corey exponent (n_w = 2.8). This induces longer late-time storage dynamics (the proposed model over-estimates storage at late time for the green curves). However, most of the transient dynamic is captured even for these cases. The model of Di Donato et al. - despite improving drastically over the one proposed in Gilman (1986) - appears to underestimate the storage for all the cases. We believe this is directly linked to neglecting the viscous timescale (first term of Equation (3.3)), as the work of Di Donato et al. was focused on oil-gas systems, assuming $\mu_w \gg \mu_n$. The proposed model shows smaller errors for all cases when compared to the work of Di Donato et al.. For all cases, Gilman (1986) over-estimates the transfer rate by several orders of magnitude and fails to represent the drainage process with any accuracy. This error is particularly high for match stick geometries, where the time to reach maximum recovery shows a difference that can reach the order of magnitude of a thousand days when compared to the high-resolution simulations. These results were also observed by Abushaikha and Gosselin (2008). The results shown in this section provide an indication that our new model can be successfully used to model CO_2 storage in NFRs.

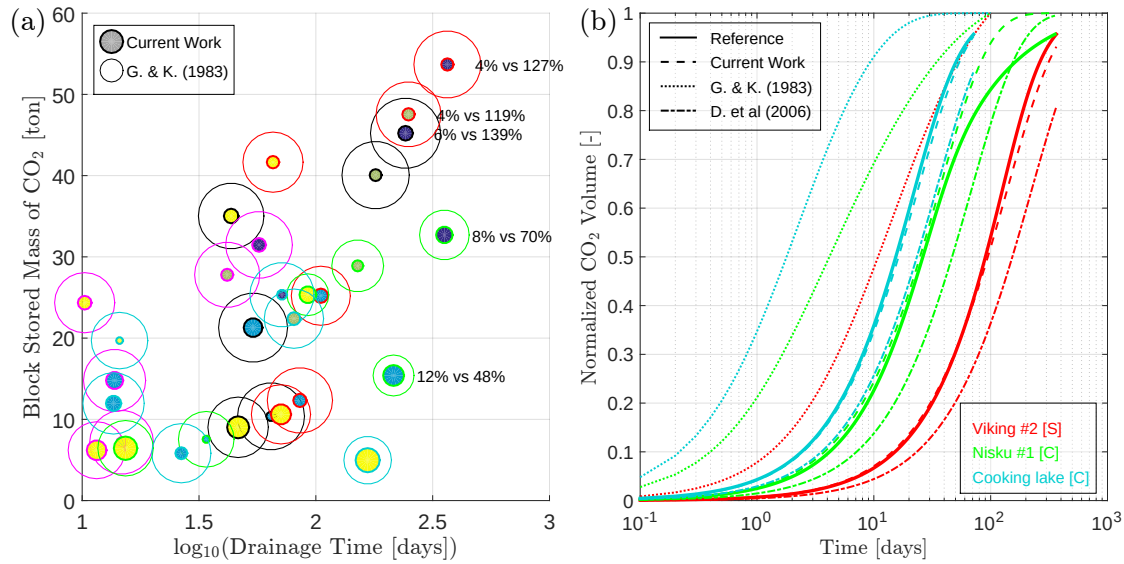

Figure 3.3: Evaluation of transfer functions for the sugar cube matrix geometry. Circles edges and fills are colored following the pattern in Figure 3.2. (a) The size of the circle is proportional to the error when using Gilman (1986) (non-solid circles) and the transfer function proposed in this chapter (solid circles) to model the drainage process, relative to the 3D numerical model. (b) Representative drainage curves showing the reference solution, given by high-resolution simulations of a single block (solid lines), the Gilman (1986) transfer function (dotted line), the Di Donato et al. (2006) transfer function (dash-dotted line) and the transfer function proposed in this chapter.

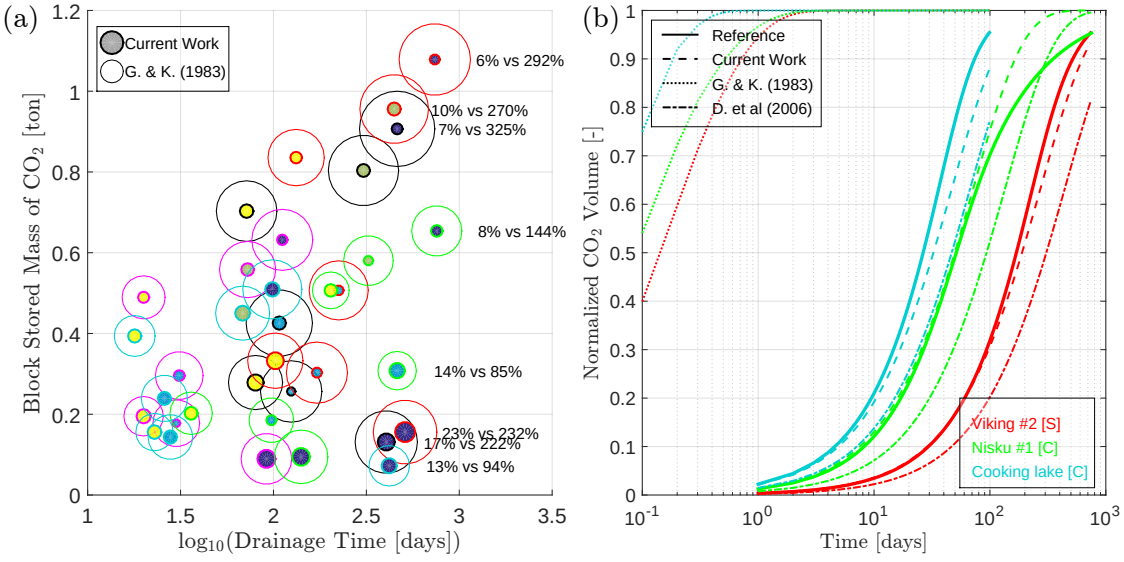

Figure 3.4: Evaluation of transfer functions for the match stick matrix geometry. Circles edges and fills are colored following the pattern in Figure 3.2. (a) The size of the circle is proportional to the error when using Gilman (1986) (non-solid circles) and the transfer function proposed in this chapter (solid circles) to model the drainage process, relative to the 3D numerical model. (b) Representative drainage curves showing the reference solution, given by high-resolution simulations of a single block (solid lines), the Gilman (1986) transfer function (dotted line), the Di Donato et al. (2006) transfer function (dash-dotted line) and the transfer function proposed in this chapter.

The previous sections have focused on the storage of CO_2 at the scale of a matrix block. The negative impact of fractures on storage is summarized in Figure 3.2(b), which shows the percentage of volume (or mass, as we consider incompressibility of CO_2 in supercritical state) that is lost when a system of interconnected fractures is present in the formation. This figure shows that fractures can reduce the CO_2 storage in a single block to values as low as 5% of the effective pore volume. This happens because a matrix block surrounded by CO_2 will be drained just until capillary and gravity forces balance. There are cases, however, in which good storage performance can be achieved, particularly for low capillary pressure levels and for warm basins. This static analysis show that a reasonable amount of CO_2 may be stored in a fractured reservoir depending on the geological environment.

Another aspect of concern regarding CO_2 storage in NFRs is the fate of the CO_2 plume. Since fractures have typically conductivities that are orders of magnitude higher than that of the rock matrix, it is usually speculated whether injected CO_2 may quickly flow through the fracture system without a significant volume of it being transferred to the rock matrix. We investigate the impact of fractures at the field scale by running two-dimensional simulations of CO_2 injection in a conceptual anticline geometry. We first compare the results of injection of a certain amount of CO_2 in a unfractured formation with the injection of the same amount in a formation that has the same properties of the rock matrix, but shows the presence of an interconnected fracture system. We then investigate why and how CO_2 injection rates should be limited in order to avoid the spilling of CO_2 through the fracture system without allowing for transfer to the matrix.

3.5.1 *Setup of the numerical simulation experiments*

The numerical experiments shown hereafter correspond to CO_2 injection in the geometry representing a symmetric anticline, as shown in in Figure 3.5. The anticline dips at $\approx 2.8\ ^o$ (25 m/500 m). A well that is completed over the entire thickness of the formation (L_z = 100 m) injects CO_2 at rates that will be specified for each experiment. No-flow boundary conditions are considered at the top and bottom of the model, representing a cap rock and an underlying non-permeable formation. We consider the lateral boundaries open for flow and at hydrostatic pressure of the resident fluid, which is brine. Simulations are carried out using MRST. The corner-point grid structure is used to discretize the domain. N_x = 101, N_y = 1 and N_z = 100 blocks are used for all the simulations (the extra block in x-direction is used to ensure symmetry of the grid).

The fracture system is represented as a homogeneous medium with no capillary pressure and each fracture cell has a porosity of $\phi_f = 0.01$ (Firoozabadi and Thomas, 1990). No residual saturations and quadratic relative permeabilities are assumed for the fracture system (Rossen and Kumar, 1992). The absolute permeability of the fracture system is set to be anisotropic: $k_{fx} = 405$ mD and $k_{fz} = 4053$ mD, corresponding to a nearly vertical set of fractures. These are realistic values taken from the In-Salah Gas Joint Venture CO_2 storage project (Cavanagh and Ringrose, 2011) and were found via discrete fracture network (DFN) modelling (Iding and Ringrose, 2010). Four cases are considered in the forthcoming analysis. Two cases with high drainage times (viking#2-shallow-cold-low-P_c and cardium#2-deep-warm-low-P_c) and two further cases with low drainage times (nisku#1-shallow-cold-low-P_c and cooking-lake-shallow-warm-high-P_c). These cases define the rock properties for the unfractured simulations and β and S_{nm}^{max} for the dual-porosity simulations (see Table 3.2 for a summary of the selected cases).

3.5.2 *Impact of fractures in the field-scale storage of CO_2 in a fractured anticline*

Figure 3.6 shows a comparison of CO_2 storage in fractured and unfractured formations. As the existence of a fracture system may not be known in advance, we aim at understanding how an operation that was designed for a unfractured formation would be affected by the presence of fractures. This situation occurred in the In Salah Project, where the existence of a fracture system was not recognized before the injection started (Iding and Ringrose, 2010). Hence, we compare the results of CO_2 injection in a fractured formation, using the dual-porosity model developed in Section 3.3.1, with the injection in an unfractured formation. In our simulations, a volume equal to the effective pore volume available for safe storage on the structural trap, PV^{eff}, (see Figure 3.5) is injected over 1000 days. After this time, injection ceases as further injection would cause CO_2 to spill through the boundaries, even without the presence of fractures.

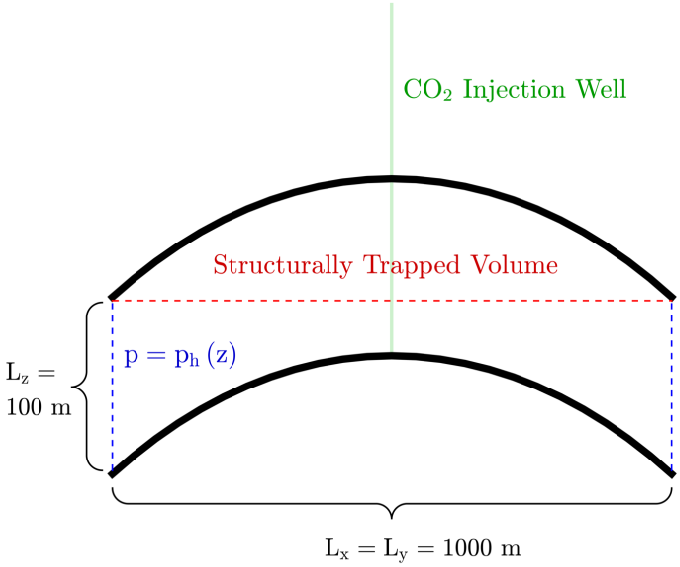

Figure 3.5: Conceptual picture of the anticline geometry considered for the simulations in Section 3.5. CO_2 is injected on a well completed over the entire thickness of the anticline (green line). Injected CO_2 is structurally trapped only in the upper portion of the anticline, between the dashed red line and the cap rock. Lateral faces are open for flow, and hydrostatic pressure of brine is assigned to these faces. No-flow boundary conditions are considered for the top and bottom boundaries, representing the overburden and underburden to the aquifer. Note that the picture is vertically exaggerated and the dip of the anticline is not to scale.

Figure 3.6(a) and Figure 3.6(b) show the CO_2 saturation fields at t = 1000 days before injection ceases for the fractured and unfractured simulations, respectively. A well-developed plume is seen in the unfractured simulation, while in the fractured case, the presence of a high-permeability fracture system induces the quick segregation of CO_2 at the top of the fractures. The transfer to the matrix takes time. The time-scale is determined by the transfer rate coefficient β (see Equation (3.10)). Figure 3.6(c) shows the mass of CO_2 in the system for the fractured and unfractured models. We note a decrease of $\approx$ 27% in the stored CO_2 volume when fractures are added to the system. If the CO_2 that is stored in the matrix (dashed line) is compared to the case where the fractures are absent (solid blue line), the difference ($\approx$ 32%) corresponds precisely to $100\% - V^{\%}_{CO_2}$ (see Section 3.2.2 and Figure 3.2(b)). Therefore, since most of the CO_2 is stored in the matrix, $V^{\%}_{CO_2}$ may be used as a correction factor for volumetric CO_2 storage estimates, at least as a first-order approximation. This is more evident in Figure 3.7, where we plot the injected volume normalized by the effective pore volume (PV^{eff}). CO_2 in the matrix (dash-dotted lines) match $V^{\%}_{CO_2}$ approximately, as explained above. The difference between the solid curves and the thick gray line represents the amount of CO_2 that left the system through the boundaries, i.e. spilling the anticline through the top corners of the model. An ideal operation would ensure that all the injected CO_2 remains in the domain, where it can be

safely trapped. The spill times depend not only on ρ_α, injection rate and thickness of the formation (which determines the shape of the plume in the fracture system) but also on the transfer rate β and S_{nm}^{max}. Compare, for instance, the viking#2 and nisku#1 cases, that correspond to the same environment (shallow-cold). If there was no transfer, the stored CO_2 for these two cases should be the same, as the rock and fluid properties would be the same. However, because the transfer dynamics is faster for the nisku#1 case (see Figure 3.3(b)), injected CO_2 is transfered faster to the matrix, avoiding the early spill time seen in the viking#2 case. There is, in fact, an optimal injection rate that avoids spilling if the correct amount of CO_2 is injected for each case. This topic is approached in the next section.

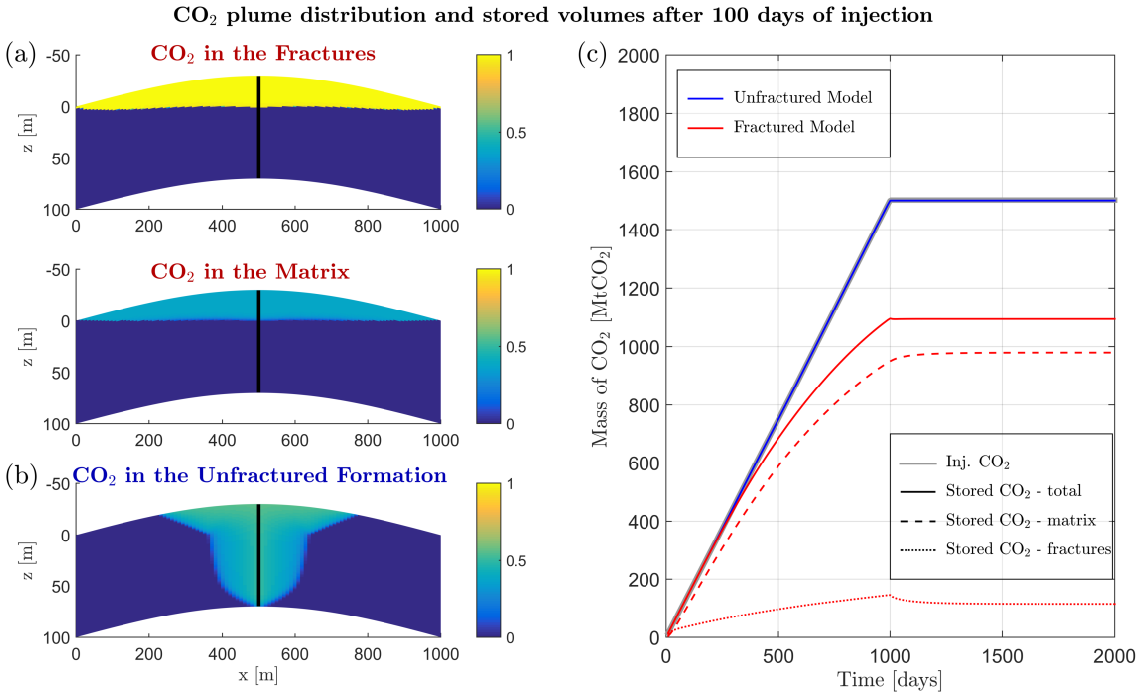

Figure 3.6: CO_2 saturation distributions for the viking#2-shallow-cold-low-P_c properties, for (a) the fractured anticline (top fractures, bottom matrix) and (b) the unfractured anticline. (c) Injected and stored mass of CO_2 for both simulation cases. Storage efficiency is reduced in the fractured case, and CO_2 escapes through the boundaries (solid red line diverging from thick gray line at $\approx$ 400 days), reflecting the fast flow of CO_2 through the fracture system.

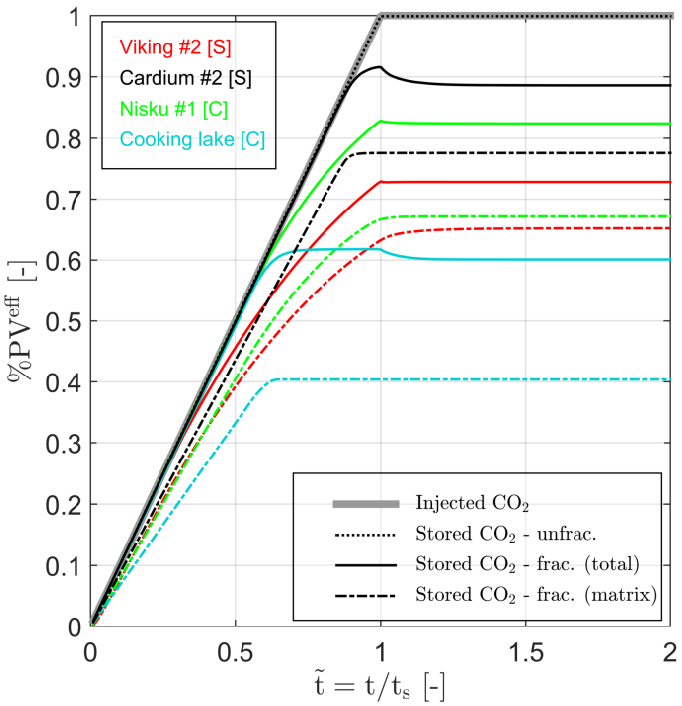

Figure 3.7: Relative injected and stored volumes for the field-scale simulations. Time is normalized by 1000 days, when injection stops. Solid lines represent the total volume of CO_2 stored in the fractured anticlines, while dotted lines represent the total volume of CO_2 stored in the corresponding unfractured model. The difference between both curves shows the reduction in storage efficiency loss when a system of connected fractures exists in the formation.

3.5.3 *Injection rate considerations*

We now explore aspects regarding injectivity of CO_2 in NFRs. We disregard the injection rate limitations imposed by the integrity of the cap rock in this analysis, noting that the fractures will allow for higher injectivity compared to an unfractured formation. We now inject the maximum amount of CO_2 that the matrix can store for each case, which is given by $V_{inj} = S_{nm}^{max} \phi_m V_a$, where V_a is the volume of the anticline (the volume of the formation above the red dashed line in Figure 3.5). We consider three different injection rates: $V_{inj}/(1000$ days), $V_{inj}/(500$ days) and $V_{inj}/(100$ days). The results are summarized in Figure 3.8. We focus the discussion, initially, on Figure 3.8(a). The difference between injected CO_2 (solid line) and stored CO_2 (dashed line) is the amount of CO_2 that spilled. We note that all three initial injection rates (blue, red and black lines) are high enough so that CO_2 travels quickly through the fracture system without allowing for enough time to transfer to the matrix. If the injection rate is reduced to $V_{inj}/(1600$ days) (magenta curve), the injected CO_2 is safely trapped in the anticline. The same procedure is applied to the other three cases to reveal the injection rates that minimize the spilled CO_2. As mentioned in the previous section, one must be more conservative regarding the injection rate for the low

β cases (viking#2 and nisku#1). Ideal injection rates are, in fact, monotonically increasing with β.

The following simple model serves as an estimation of the maximum injection rate in NFRs:

$$q_{inj}^{max} = \frac{V_{inj}}{t_{95}}. \tag{3.14}$$

This model implies that one should inject at a rate that is no higher than the maximum amount one can store in the rock matrix divided by the time it takes to saturate 95% of the matrix pore volume. t_{95} may be easily calculated by inverting Equation 3.10 and evaluating it for $V_{CO_2}(t)/V_{CO_2}^{max} = 0.95$. Although this model has some physical reasoning behind it, it lacks a fundamental component for a more accurate estimate, which is the geometrical component of the CO_2 plume in the fracture system. Table 3.2 shows a summary of the injection rates found via numerical simulation and using the model given by Equation (3.14). We define the accuracy factor of the injection rate estimate as q_{inj}^{max}/q_{num}, where q_{num} is the optimal injection rate obtained from numerical simulations (magenta curves in Figure 3.8). The accuracy factor ranges from ≈ 4 to ≈ 7. This model, despite being simple, provides an order-of-magnitude estimate of the optimal injection rate. It should be noted that the accuracy factors are overestimated with the 2D numerical simulations carried out in this section. The consideration of a third dimension in the y direction would lead to a higher q_{num}, making this value closer to the proposed model given by Equation 3.14. This gives us indications that this model could serve as a good initial indicator of the magnitude of the injection rate that avoids the early-spill of CO_2 for anticlines.

Case	β [1/s]	S_{nm}^{max} [-]	q_{num} [m^3/day] (ideal value - numerical)	q_{inj}^{max} [m^3/day] (Equation 3.14)	q_{inj}^{max}/q_{num}
viking#2	8.48×10^{-8}	0.39	894	3499	≈ 4
nisku#1	3.22×10^{-7}	0.45	1158	8068	≈ 7
cooking-lake	1.99×10^{-6}	0.21	5406	23270	≈ 4
cardium#2	6.03×10^{-7}	0.45	4634	24175	≈ 5

Table 3.2: Summary of the selected cases for the field-scale simulations.

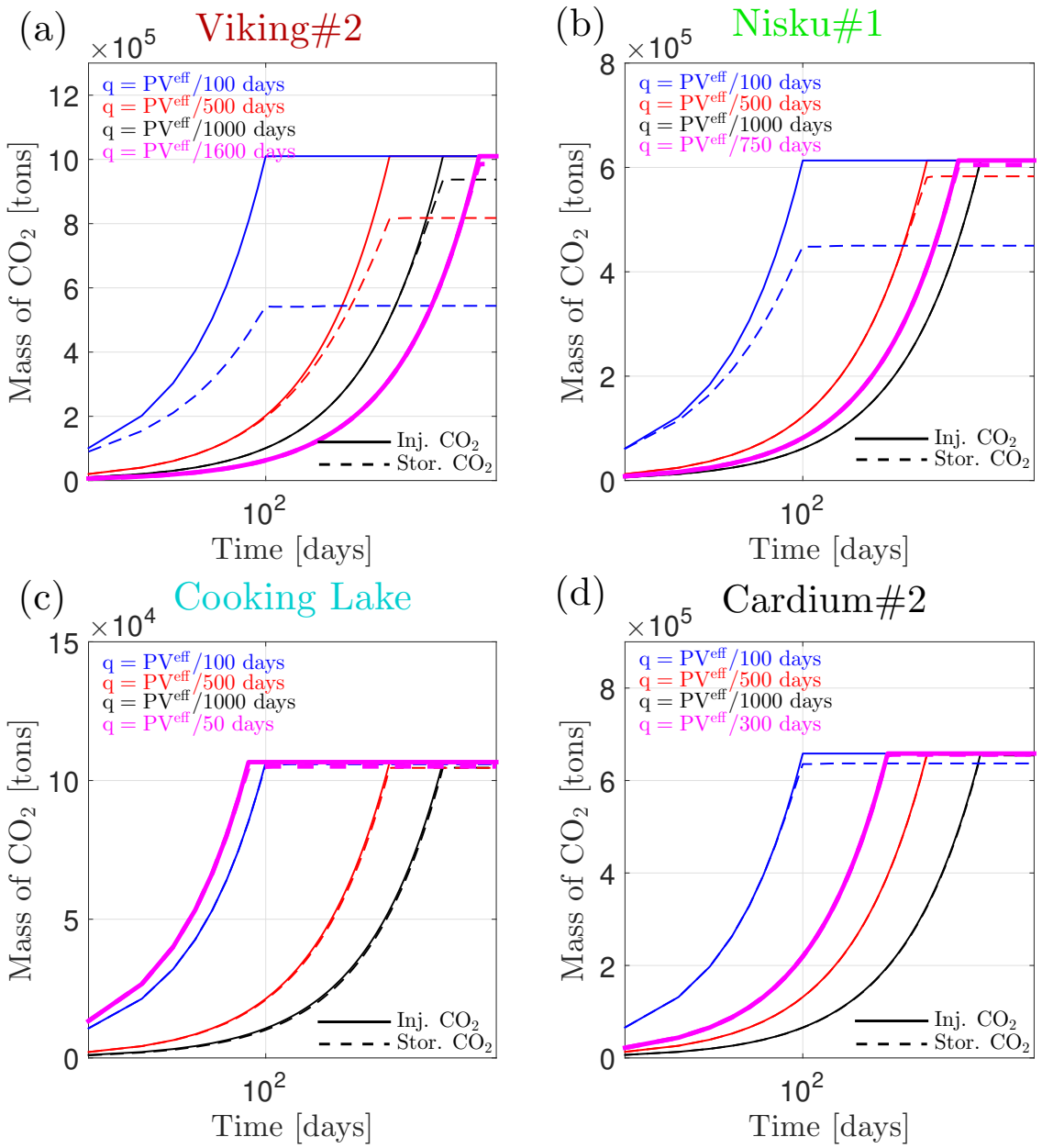

Figure 3.8: Injected (solid lines) and stored mass (dashed lines) of CO_2 for (a) viking#2, (b) nisku#1, (c) cooking-lake and (d) cardium#2 cases in the geometry depicted in Figure 3.5. Each case has its "optimal" injection rate, i.e. an injection rate that minimizes spilling of CO_2.

3.6 SUMMARY AND CONCLUSIONS

Gravity drainage is the physical mechanism that drives CO_2 into the matrix during CO_2 injection in NFR's. Drainage is mainly driven by buoyancy that is caused by the density difference between CO_2 and brine phases and opposed by capillary forces. Despite being an important transfer mechanism, currently existing transfer functions to capture this phenomenon fail to represent the physics accurately and are typically designed for oil and gas applications. Therefore, one of the key contributions of this chapter was the development of a new transfer func-

tion that models CO_2-induced gravity drainage accurately. This transfer function is based on a timescale analysis of the physics of immiscible multi-phase flow combined with an exponential model, as suggested initially by Di Donato et al. (2006). A more formal treatment of each timescale term, combined with the inclusion of the viscous term, improves the original model of Di Donato et al. and the widely used model of Gilman (1986). Our new model requires no fitting to numerical simulations, is consistent and captures the drainage process accurately. The transfer function presented here is easy to implement in any dual-porosity simulator.

At the matrix-block scale, we show that the percentage of the matrix-block volume that is effectively used for storage, is higher for deep formations in cold basins and lower for shallow formations in warm basins ($\approx$ 40% vs $\approx$ 15%). In contrast to this result, deep formations in cold basins will lead to a higher mass of CO_2 stored per unit volume, due to the high density of the supercritical fluid. This indicates a competing behavior of CO_2 density in fractured reservoirs: for a fixed brine density ρ_b, a high value of ρ_c increases the stored mass per unit volume, but decreases the stored mass on each block due to the decreased density difference $\Delta\rho$. Our studies show that in a shallow formation in a warm basin the negative impact of fractures on the storativity of CO_2 is less pronounced. Therefore, in such formations, the existence of an interconnected system of fractures would not grossly impact the final storage.

At the field scale, we have shown that the injection rate for a fractured anticline must be designed to avoid what is termed here as “early spill": the fast flow of CO_2 through the fracture system without significant storage in the matrix. We have shown that the spill time for a conceptual anticline geometry is heavily dependent on the transfer rate between the two continua. An initial model to estimate the order of magnitude of the ideal injection rate is proposed (Equation 3.14). Numerical experiments shown in Figure 3.8 indicate that this model captures the order of magnitude of the optimal injection rate that avoids early spill, even though it does not consider the geometrical aspect of the CO_2 plume in the fracture system. Although we understand that the major injection rate design criteria for CO_2 storage is still the integrity of the sealing formation, we point out in this chapter that a new aspect of injection rate determination must be taken into consideration when using an NFR for storage.

4

MODELLING OF CAPILLARY-DRIVEN SPONTANEOUS IMBIBITION

4.1 INTRODUCTION

The spontaneous invasion of a wetting phase into a porous medium due to capillary forces is a remarkable physical phenomenon relevant to a wide range of geological and engineering applications. Most important to the context of this thesis, spontaneous imbibition (SI) is one of the main mechanisms of fluid exchange between fractures and matrix during geological storage of CO_2 in NFR's. As seen in Chapter 1, after injection of CO_2 stops, the CO_2 plume migrates upwards and accumulates at the cap-rock. This generates an upwards flow of brine coming from the bottom of the aquifer to saturate the space left by the CO_2 in the fractures. Matrix blocks that were previously drained by CO_2 are then surrounded by brine, that shall displace CO_2 back to the fractures due to capillary forces. Hence, understanding SI is of great importance to understand the imbibition of brine into CO_2 saturated rocks in geological storage of carbon dioxide (Nordbotten and Celia, 2012).

Counter-current spontaneous imbibition is mathematically described by a nonlinear diffusion equation. Finding analytical solutions that are valid at early and late times has been an open challenge for many years. At early time, that is, before the advance of the wetting phase front is influenced by a no-flow boundary condition, the cumulative imbibed volume scales with $\sqrt{t}$ (Lucas, 1918; Washburn, 1921). The late-time behaviour, on the other hand, is characterized by a decrease in the imbibition rate. It is usually presumed to follow approximately an exponential expression of the form $V/V_\infty = 1 - e^{-\lambda \cdot t}$ (Aronofsky et al., 1958), where λ describes the rate of the transfer process. Models for this parameter have been proposed over the last century (Washburn, 1921; Li and Horne, 2006; Zhou et al., 2002; Ma et al., 1997; Mattax and Kyte, 1962; Tavassoli et al., 2005), but only recently a general scaling group based on the analytical solution for the counter-current spontaneous imbibition in semi-infinite domain was developed (Schmid and Geiger, 2012, 2013). This group properly includes the effects of rock wettability, viscosity ratio and other physical parameters, such as arbitrary capil-

lary pressure and relative permeability curves, and provides a good agreement with a large body of experimental data of spontaneous imbibition.

In the context of modeling and simulation of fractured reservoirs, dual-porosity models provide a framework for simulation of such geological formations by considering the fracture network as a second porous medium/continuum that is superposed to the matrix rock (Warren and Root, 1963). The fluid interchange between the two continua is modelled by means of a transfer-rate function. In this sense, the exponential model for cumulative imbibition is equivalent to a first-order model where the transfer rate depends linearly on the saturation in the matrix block (Di Donato et al., 2007). The first order transfer function has been widely used as a model to represent the whole imbibition process. It is observed, however, that this model underestimates drastically the imbibed volume at the onset of the process (Zimmerman et al., 1993; Geiger et al., 2013; Tecklenburg et al., 2013, 2016).

Based on a single-phase development (Zimmerman et al., 1993), in Zimmerman et al. (1996), an approximate solution for the pressure response of a spherical block subjected to a step-function signal was presented. Lu et al. (Lu et al., 2008) suggested a similar first-order model with a boost or correction factor to accurately model the early-time behaviour. Although these approaches represent a significant improvement over classical first-order transfer-functions, they lack generality since they rely on lumped models that average the saturation in a matrix block. However, the exchange of fluids between fracture and matrix heavily depends on the specific shape of the saturation distribution and hence cannot generally be represented by a single average quantity. Therefore, these models lack generality and require specific fitting to each set of physical parameters. In an alternative approach, Geiger et al. (2013); Tecklenburg et al. (2013, 2016) linearise and solve the non-linear diffusion equation for SI in Laplace space to obtain an exact solution for the transfer-rate term. Nevertheless, linearisation of the diffusion coefficient may lead to large errors in the estimate of the imbibition rate.

At early time, the $\sqrt{t}$ scaling can be used to develop integral solutions to the non linear diffusion equation that models counter-current spontaneous imbibition. In particular, we consider the solutions presented in McWhorter and Sunada (1990), Schmid et al. (2011) and Doster et al. (2012), as they do not assume any particular model for capillary pressure and relative permeabilities. This represents a general solution for the SI problem. At early time we are able to calculate the imbibed volume exactly for the one-dimensional case. At late time, the aforementioned model of Aronofsky et al. (1958) with the universal transfer rate coefficient is used. Combining both solutions requires a proper characterization of the transition between the two regimes. The idea of combin-

ing an early-time expression based on semi-analytical solution with a late-time expression based on the exponential approximation was suggested initially in Chen et al. (1995) and later in Tavassoli et al. (2005). In both cases it is assumed that the transition between early and late time occurs when the semi-analytical solution ceases to be valid. That is, when the wetting phase front reaches the no-flow boundary of the domain. However, here we show that the imbibition scales proportional to $\sqrt{t}$ for a significantly longer period.

We therefore present in this chapter a new hybrid approach to model the imbibed volume that combines the models for early- and late-time behaviour. The early-time behaviour is represented by a semi-analytical solution and the late-time behaviour uses an approximate exponential model with a universal transfer rate constant. We specifically consider the counter-current imbibition of a wetting phase into a rock matrix at residual wetting phase saturation where symmetry allows for a one-dimensional representation. A novel method that allows us to reconstruct the saturation profile inside the physical domain and to estimate the transition time is presented. The profiles of the imbibed volumes generated with this hybrid model are evaluated by comparing the results with a series of high-resolution numerical simulations with realistic physical properties arising from oil recovery and CO_2 storage applications. Additionally, an analysis of the sensitivity of the results with respect to the viscosity ratio and the capillary pressure is presented.

The chapter is structured as follows. In Section 4.2 the governing equations, as well as a general discussion of the imbibition process and previously published early- and late-time imbibition models are given. In Section 4.3, we present a model that combines both early- and late-time solutions and provides a physics based method to estimate the transition between the two regimes. Section 4.4 shows simulations using realistic parameters to illustrate the validity of our approach for different physical systems. Finally, in Section 4.5, the conclusions are discussed.

4.2 Models for counter-current spontaneous imbibition

4.2.1 *Governing Equations*

The dynamics of two-phase flow in a porous medium are described by a set of mass conservation equations that were introduced in section 1.2 of Chapter 1. Here we consider the problem of a one-dimensional homogeneous porous medium at residual wetting phase saturation that is sealed at one side and exposed to the wetting phase on the opposite boundary (Figure 4.1).

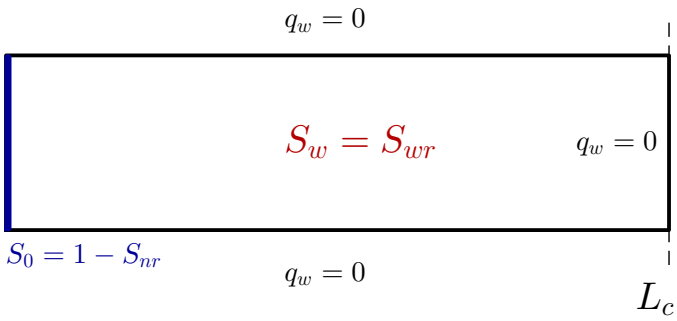

Figure 4.1: Sketch of the one-dimensional domain with initial and boundary conditions.

The domain is spontaneously imbibed by the wetting phase, leading to counter-current flow of the non-wetting phase. Since capillary pressure is generally not strong enough to compress the fluids by itself, we treat the phases as incompressible. Therefore, Equation (1.7) may be employed to model SI in this context. The counter-current flow assumption implies that the total flow of phases is zero everywhere ($q_t = 0$). This allows us to simplify the fractional flow equations and write a single non-linear equation to model SI:

$$\phi \frac{\partial S_w}{\partial t} - \frac{\partial}{\partial x}\left\{ D_w\left(S_w\right) \frac{\partial S_w}{\partial x} \right\} = 0, \tag{4.1}$$

where $D_w\left(S_w\right)$ denotes the capillary diffusion coefficient defined as

$$D_w\left(S_w\right) \equiv -f_w k \lambda_n \frac{dp_c}{dS_w}. \tag{4.2}$$

The following set of initial and boundary conditions is considered:

$$\begin{aligned} &S_w\left(x, t = 0\right) = S_{wr}, \\ &S_w\left(x = 0, t\right) = S_0, \\ &\left.\frac{\partial S_w}{\partial x}\right|_{L_c} = 0, \end{aligned} \tag{4.3}$$

where S_0 denotes the saturation at the boundary, L_c is the size of the physical domain and S_{wr} is the residual wetting phase saturation. Equation (4.1) together with conditions given by Equation (4.3) and a parametrization for the capillary pressure $p_c\left(S_w\right)$ form a closed set of equations that describe one-dimensional counter-current spontaneous imbibition. The equivalent dimensionless form of Equation (4.1) is given by

$$\frac{\partial S_w}{\partial \tilde{t}} - \frac{\partial}{\partial \tilde{x}}\left\{ \tilde{D}\left(S_w\right) \frac{\partial S_w}{\partial \tilde{x}} \right\} = 0, \tag{4.4}$$

with the dimensionless diffusion coefficient given by

$$\tilde{D}\left(S_w\right) = -\frac{1}{\phi} f_w k \lambda_n p_e \frac{T}{L_c^2} \frac{d\tilde{p}}{dS_w}, \tag{4.5}$$

where $\tilde{t} = t/T$, $\tilde{x} = x/L_c$ and $\tilde{p} = p_c/p_e$. p_e is the capillary entry pressure and T a characteristic timescale of imbibition. Note that the physical problem is completely determined by a single dimensionless group and the shape of the constitutive functions $k_{rw}(S_w)$, $k_{rn}(S_w)$ and $p_c(S_w)$. We can now use the dimensionless groups to map solutions from different problems by simply adjusting the timescale T. In the remainder of this chapter we choose particular sets of parameters for the sake of concreteness but note that the findings are general.

4.2.2 *Behaviour of the Imbibed Volume*

Semi-analytical solutions exist for the mathematical system described by Equations (4.1) and (4.3) when $L_c \to \infty$ (see e.g. McWhorter and Sunada (1990)). Recall that at early time, the diffusive nature of imbibition leads to a $\sqrt{t}$ scaling of the imbibed volume. Once the boundary at L_c has a significant impact on the overall behaviour, the rate of imbibition slows down. The scaling of this process reflects the whole non-linear nature of spontaneous imbibition into a finite domain and a universal scaling is not available. The commonly applied exponential expression proposed by Aronofsky et al. (1958) captures the process reasonably well at late time, but not at early time as discussed before (Zimmerman et al., 1993, 1996; Geiger et al., 2013; Tecklenburg et al., 2013, 2016). Figure 4.2 illustrates the temporal evolution of the wetting phase during SI. At the onset of the process, the imbibed volume increases proportional to $\sqrt{t}$, as predicted by Lucas (1918). The semi-analytical self-similar solution using this scaling behaviour matches perfectly the simulation results. Once the imbibition slows down, this solution fails to capture the temporal evolution of the imbibition and the curve is now approximately exponential. Conversely, the sigmoidal function from the exponential model fails to represent the temporal evolution of the early time. This period can be long for many applications. An accurate estimate of the time when the transition from early- to late-time behaviour occurs, based solely on the physical parameters of the problem, forms the main contribution of this manuscript and is presented in the following sections.

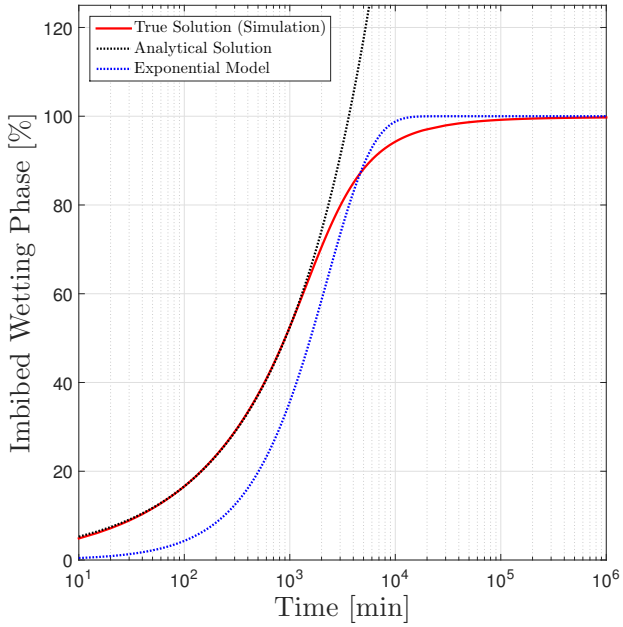

Figure 4.2: Semi-logarithmic plot of the imbibed volume vs time for an illustrative example.

4.2.3 *Early Time Regime*

The wetting phase volume that has imbibed into the porous medium is given by the solution of the initial and boundary value problem defined in Equations (4.1-4.3). This non-linear parabolic equation has no general solution and requires some numerical solution techniques (Schmid et al., 2011; Bjornara and Mathias, 2013). However, at the onset of imbibition a distinctive front with a finite velocity propagates into the medium due to the vanishing diffusion coefficient for $\lim_{S \to S_r} D(S) = 0$. Hence, until the saturation front reaches the boundary at L_c, the solution is identical with an imbibition into a semi-infinite domain.

A semi-analytical solution for this problem was developed in McWhorter and Sunada (1990) where the imbibition rate decreases $\propto t^{-1/2}$. The proportionality constant is obtained by solving an implicit integral equation numerically. For convenience we briefly present the solution below. Schmid et al. (2011) have shown that the counter-current imbibition problem is well-posed once the saturation in the boundary is prescribed, as in Equation (4.3), i.e. there is no need to prescribe a flux boundary condition for the wetting phase. In other words, the flux in the boundary is a natural consequence of the saturation difference between the boundary and the medium.

The analytical solution is based on the fact that before the wetting phase front reaches a no-flow boundary or another wetting-phase saturation front traveling in opposite direction, the nonlinear diffusion equation shows self-similarity for the variable $\eta = xt^{-1/2}$. The problem can then be rewritten as a nonlinear first-

order differential equation, which is solved using an iterative procedure. The wetting phase flux in the left boundary is naturally imposed by the saturation at this point, S_0, and is defined as

$$q_w(0,t) = At^{-1/2}, \tag{4.6}$$

where A is the imbibition rate constant, with dimension of $\left[L \cdot t^{-1/2}\right]$. This parameter is related to the saturation in the boundary through the following expression

$$A^2 = \frac{\phi}{2} \int_{S_{wr}}^{S_0} \frac{(S_w - S_{wr}) D_w(S_w)}{F_w(S_w)} dS_w, \tag{4.7}$$

where F_w is the general fractional flow function defined as

$$F_w(x,t) = \frac{q_w(x,t)}{q_w(0,t)}. \tag{4.8}$$

We obtain the fractional flow function and its derivative by solving iteratively the following nonlinear integral equations

$$\begin{aligned} F_w(x,t) &= 1 - \left[\int_{S_w}^{S_0} \frac{\left(S_w' - S_w\right) D_w\left(S_w'\right)}{F_w\left(S_w'\right)} dS_w'\right] \cdot \left[\int_{S_{wr}}^{S_0} \frac{\left(S_w' - S_{wr}\right) D_w\left(S_w'\right)}{F_w\left(S_w'\right)} dS_w'\right]^{-1}, \\ F_w'(x,t) &= \left[\int_{S_w}^{S_0} \frac{D_w\left(S_w'\right)}{F_w\left(S_w'\right)} dS_w'\right] \cdot \left[\int_{S_{wr}}^{S_0} \frac{\left(S_w' - S_{wr}\right) D_w\left(S_w'\right)}{F_w\left(S_w'\right)} dS_w'\right]^{-1}, \end{aligned} \tag{4.9}$$

which allows us to reconstruct the saturation front through the following expression

$$x(S_w,t) = \frac{2A}{\phi} F_w'(S_w)\, t^{1/2} = \frac{Q_w(t)}{\phi} F_w'(S_w), \tag{4.10}$$

where $Q_w(t)$ is the cumulative imbibed volume until time t. We note that this expression is the capillary analogue for the well known Buckley-Leverett equation (Buckley and Leverett, 1942). Equations (4.7), (4.9) and (4.10) form a closed set of nonlinear integral equations that can be solved to obtain the saturation profile and cumulative imbibed volume at each time step. Strictly speaking, due to the assumption of an infinite domain, this equation is only valid before the saturation front reaches the closest no-flow boundary. This time is denoted here as t^* and is given by setting $x(S_w,t) = L_c$ in Equation (4.10):

$$t^* = \left(\frac{L_c \phi}{2A F_w'(S_{wr})}\right)^2. \tag{4.11}$$

The relative imbibed volume for the early-time regime $E(t)$ is calculated as the ratio of the imbibed wetting phase volume and the maximum volume available for imbibition

$$E(t) \equiv \frac{Q_w(t)}{(1 - S_{wr} - S_{nr})\,\phi L_c} = \frac{2At^{1/2}}{(1 - S_{wr} - S_{nr})\,\phi L_c}. \tag{4.12}$$

Equation (4.12) is a direct result of the integration of the flow rate in the boundary over time, and it is equivalent to Equation (9) of Schmid and Geiger (2012). This solution is exact up to the tolerance at which the nonlinear problem given by Equation (4.9) is solved numerically.

4.2.4 *Late Time Regime*

As mentioned before, the Aronofsky et al. (1958) expression for the late-time regime is given by

$$\frac{V(t)}{V_\infty} = 1 - e^{-\lambda t}, \tag{4.13}$$

where the parameter λ controls the rate of imbibition. The exponential model is a proven mathematical solution for the limit of late times when the diffusion coefficient is linear. Its application to the nonlinear diffusion case, however, is a purely empirical model based on the observed behaviour of the imbibition process (Kazemi et al., 1992). The previously described analytical solution can be used to derive a universal transfer rate coefficient (Schmid and Geiger, 2012). The total imbibition time t_c is calculated as the time when the total imbibed volume (see Equation (4.6)) is equal to the recoverable volume of non-wetting phase. In other words, t_c is such as

$$\phi L_c (1 - S_{wr} - S_{nr}) = 2At_c^{1/2} \Rightarrow t_c = \left(\frac{\phi L_c (1 - S_{wr} - S_{nr})}{2A}\right)^2. \tag{4.14}$$

The transfer rate parameter is then calculated as the reciprocal of the imbibition time t_c

$$\lambda = \gamma \left(\frac{1}{t_c}\right) \cdot t \equiv \gamma \tau_c \cdot t, \tag{4.15}$$

where γ is regarded as a general fitting parameter in Schmid and Geiger (2012). The parameter γ can be understood as fixed, suitable to a wide range of rock properties, capillary and relative permeability relations, and mobility ratios without further adjustment, since the scaling group τ_c should encompass all the in-

formation of the physical system. We are then able to write and expression for the late-time behaviour of the relative imbibed volume as

$$\Lambda(t) \equiv 1 - e^{\gamma \tau_c t}. \tag{4.16}$$

We obtained $\gamma \approx 1.6$ for all the experiments carried in the course of this work.

4.3 A HYBRID MODEL FOR SPONTANEOUS IMBIBITION

The late-time expression given by Equation (4.16) is normally used to model the relative imbibed volume during SI (Di Donato et al., 2007; Lu et al., 2008; Schmid and Geiger, 2013). However, as shown in Figure 4.2, this model underestimates the rate of imbibition at early time. We hence propose an imbibition model that has the following shape:

$$\Psi(t) \equiv H\left(\hat{t} - t\right) \cdot E(t) + H\left(t - \hat{t}\right) \cdot \Lambda(t + \Delta t), \tag{4.17}$$

where $H(x)$ is the Heaviside function that vanishes for negative values and is unity for positive values. The hybrid model switches from early to late time on a predefined transition time $\hat{t}$. We select the step function to transition between the models with just one parameter. It is important to note, though, that other authors have suggested different transition kernel functions (see, e.g., Tecklenburg et al. (2013) and Tecklenburg et al. (2016)) that may smooth out the transition between both models and may replace $H(x)$ accordingly in Equation (4.17). These approaches, however, introduce additional parameters to characterize the transition interval.

The late-time model defined by Equation (4.16) is shifted by Δt in time to match the early-time solution value on $\hat{t}$ and is calculated as

$$\Delta t = -\frac{1}{\gamma \tau_c} ln\left(1 - \frac{2A\hat{t}^{1/2}}{(1 - S_{wr} - S_{nr})\,\phi L_c}\right) - \hat{t}. \tag{4.18}$$

The shifting of the late-time model ensures the continuity of the model at the transition time $\hat{t}$. The imbibed volume at early time given by Equation (4.12) is matched with the value given by Equation (4.16) at this point.

4.3.1 *The characterization of the transition time*

The magnitude of the diffusion at early time and the shape of the wetting phase saturation front are characterized by the diffusion coefficient. The vanishing nature of the diffusion coefficient at residual saturations makes the existence of

a saturation front possible, even though the physical nature of Equation (4.1) is diffusive. Previous studies that aimed to merge early- and late-time models (e.g. Tavassoli et al. (2005) and Chen et al. (1995)) propose that t^* is a good estimate for the transition from early to late time. And indeed, the scaling of the imbibed wetting phase volume starts to deviate from $\sqrt{t}$ at t^* as the assumption for Equation (4.12) no longer hold true. However, observations indicate that the imbibed wetting phase volume follows an approximate $\sqrt{t}$ behaviour well after t^* was reached. We reason that during this time the spontaneous imbibition is still predominantly driven by a saturation profile that is close to the one given by the semi-analytical solution (see Figure (4.6) on the interval $[0, 2.5]$) and that the contribution of the mismatching part of the profile is of minor importance. Once this contribution to the overall diffusion is of major relevance, the scaling will substantially deviate from $\sqrt{t}$. We denote this time as $\hat{t}$. Further, we denote $\hat{S}_w$ as the saturation at the outer boundary at the moment of transition and present below a methodology to determine the transition time $\hat{t}$ based on $\hat{S}_w$.

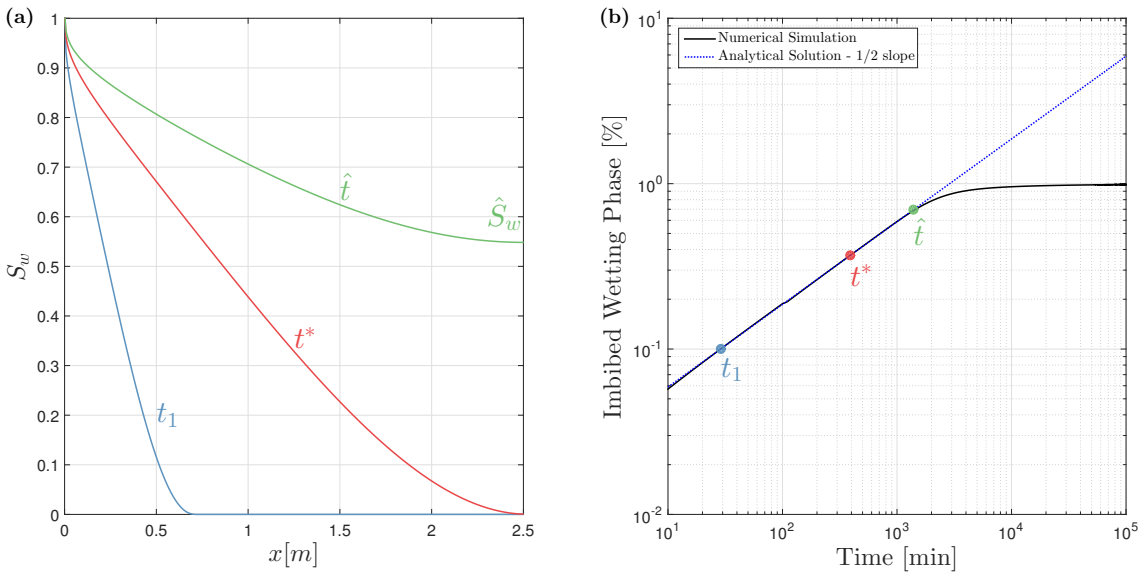

Figure 4.3: Illustrative example of a high-resolution simulation of an imbibition process into a dry domain. Figure (a) shows saturation profiles at different times and (b) a log-log plot of the imbibed volume. The instants of the three profiles are marked by filled circles. For reference the slope 1/2 corresponding to the early-time analytical solution is also shown.

4.3.2 *The choice of the transition saturation*

The value of $\hat{S}_w$, the saturation at the boundary that controls the transition to late-time behaviour, shall be chosen based on the diffusion coefficient curve. Choosing $\hat{S}_w = S_w^{max}$ may be suitable for some cases but is not necessarily applicable for all the physical situations. Diffusion curves that are skewed to the left such as the one depicted in Figure 4.4, may still have high $D_w(S_w)$ values

for saturations greater than S_w^{max}. Hence, we choose $\hat{S}_w$ as being the saturation under which some percentage of the total diffusion is achieved. Mathematically, we define a cumulative diffusion function as

$$\Delta\left(S_w\right) = \int_{S_{wr}}^{S_w} D_w\left(S_w^{'}\right) dS_w^{'}. \tag{4.19}$$

The function $\Delta\left(S_w\right)$ is invertible and we may define $\hat{S}_w$ as

$$\frac{\Delta\left(\hat{S}_w\right)}{\Delta\left(1 - S_{nr}\right)} = \xi \Rightarrow$$

$$\hat{S}_w = \Delta^{-1}\left(\xi \cdot \Delta\left(1 - S_{nr}\right)\right), \tag{4.20}$$

where $\xi \in [0, 1]$ describes the threshold of cumulative diffusion that controls the transition from early to late time (see Figure 4.5). This definition of $\hat{S}_w$ selects a transition value such that higher saturation values are less important contributions to the diffusion in the domain. Since the diffusion controls the early-time behaviour and the $\sqrt{t}$ scaling, $\hat{S}_w$ provides an adequate estimate for the saturation that controls the diffusion. The simulations shown later span a wide range of shapes and magnitudes of diffusion coefficient curves and show that the thresholds that describe the transition are typically around $\xi = 0.6$. We hence have used this level for all the simulation cases. We note that the calculation of $\hat{S}_w$ with Equation (4.20) can be easily achieved with the aid of numerical quadrature and by building a search table of $\Delta\left(S_w\right)|S_w$ values.

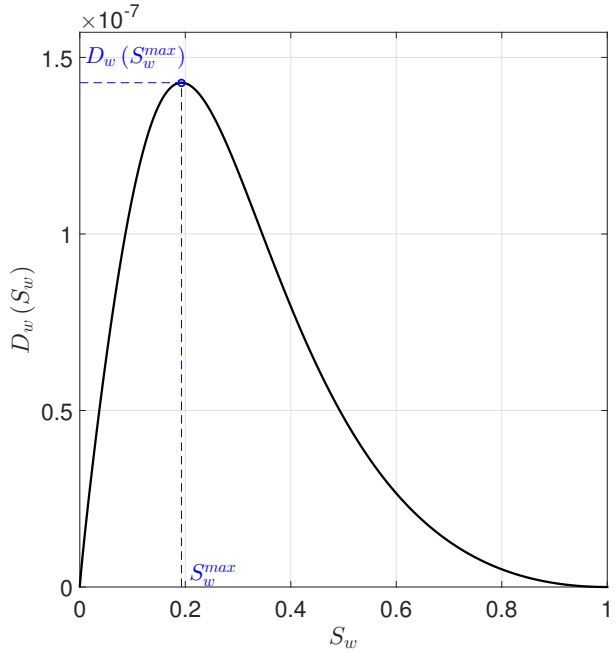

Figure 4.4: Typical shape of a diffusion coefficient curve. The maximum of the diffusion is marked by a blue circle.

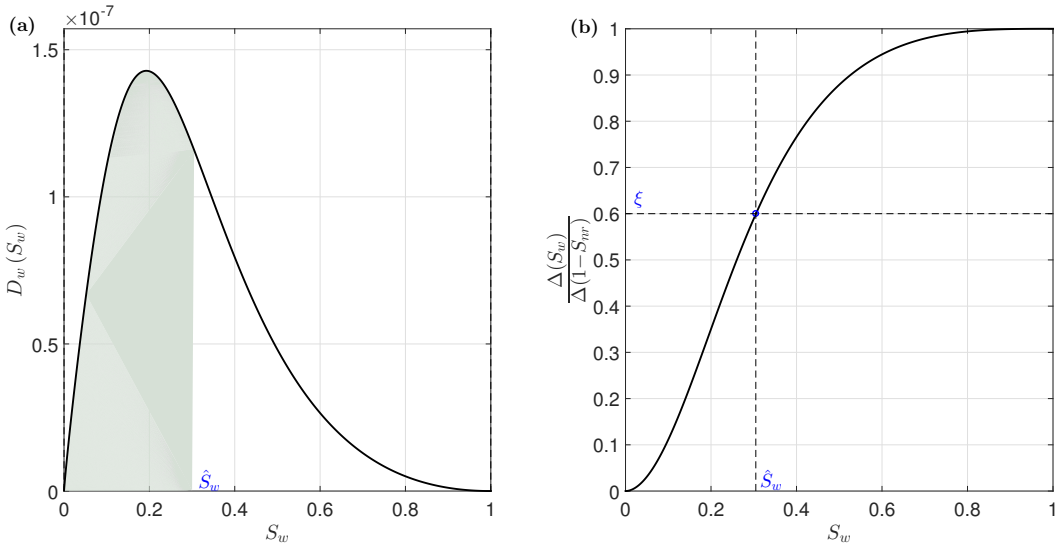

Figure 4.5: Illustration of the method to determine $\hat{S}_w$ through cumulative diffusion. Figure (*a*) shows the diffusion coefficient curve for a representative case. The shaded area marks the cumulative diffusion at $\hat{S}_w$. Figure (*b*) depicts the percentage cumulative diffusion function. We pick the saturation under which 60% of the percentage diffusion is concentrated.

4.3.3 *The calculation of the transition time - the Filling Back Procedure*

Since the analytical model is based on the solution for a semi-infinite domain, the front position defined by Equation (4.10) does not match the true solution after $t = t^*$. Consequently, one needs to reconstruct the saturation profile inside the domain to estimate the time when the saturation at L_c reaches $\hat{S}_w$. We propose a

scheme to approximate the saturation profile in the finite domain with the help of the solution for the semi-infinite domain. We take the imbibed volume that has left the finite domain according to the analytical solution and fill it back into the domain. This is done by identifying the saturation plateau $\tilde{S}_w$ that leads to an equivalent volume in the physical domain. We name this approach the *Filling Back Procedure* and formulate it mathematically as:

$$A_{L_c-\infty} = \int_{L_c}^{\infty} S_w\left(x, \tilde{t}\right) dx,$$
$$\int_0^{L_c} H\left(\tilde{S}_w - S_w\left(x, \tilde{t}\right)\right) \cdot \left(\tilde{S}_w - S_w\left(x, \tilde{t}\right)\right) dx = A_{L_c-\infty}, \tag{4.21}$$

where the Heaviside function is used to select just the part of the domain whose values of the saturation plateau are greater than the values of the saturation profile. Given a saturation profile at time $\tilde{t}$, we find the saturation level $\tilde{S}_w$ that compensates the area below the saturation curve outside the physical domain, labelled as $A_{L_c-\infty}$. With this scheme, mass is conserved and we are correcting the analytical solution profile to approximate the true profile inside the domain. In order to find $\hat{t}$, the transition time, the problem defined by Equation (4.21) is inverted. Instead of finding a saturation plateau that matches the volume at a certain time, we now seek the time $\hat{t}$ when the saturation plateau that satisfies this system is exactly $\hat{S}_w$, as defined in Equation (4.20) (see Figure 4.6). We therefore write the following nonlinear problem for $\hat{t}$:

Find $\hat{t}$ such as

$$\int_0^{L_c} H\left(\hat{S}_w - S_w\left(x, \hat{t}\right)\right) \cdot \left(\hat{S}_w - S_w\left(x, \hat{t}\right)\right) dx = \int_{L_c}^{\infty} S_w\left(x, \hat{t}\right) dx. \tag{4.22}$$

This equation can be easily solved by the bisection method. The root lies necessarily between $t = t^*$ and $t = \left(\frac{L_c \phi}{2A \cdot F'\left(\hat{S_w}\right)}\right)^2$, respectively the time when the saturation front and $\hat{S}_w$ reach L_c, according to the analytical solution. The latter is an upper limit for $\hat{t}$ because the analytical solution will always underestimate the saturation in the physical boundary L_c in comparison with the true saturation profile.

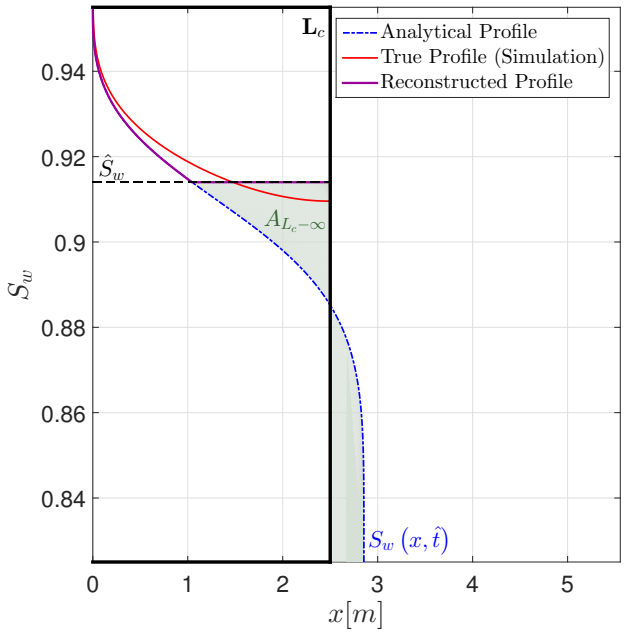

Figure 4.6: Comparison between the reconstructed saturation profile (purple curve) and the true saturation profile given by high resolution one-dimensional simulation (red curve). The analytical solution for imbibition into an infinite domain is shown as blue dash-dotted curve.The volume $A_{L_c-\infty}$ of the analytical solution that left the boundary and has been "filled back" is shown in the picture.

4.4 RESULTS AND DISCUSSION

The previous sections show a systematic procedure to predict the imbibed volume due to spontaneous imbibition for the entire duration of SI using arbitrary physical parameters. Although ξ is as a fitting parameter, it was chosen to provide a good measure of the transition time based on numerical experiments that span a wide range of possible shapes and magnitudes of the diffusion coefficient, as will be shown in this section. Hence, the parameter ξ requires no further adjustment. It is general in the sense that it may be used for any set of rock and/or fluid properties.

We now analyse the accuracy of the hybrid imbibition model through a series of high resolution numerical simulations. The first set of simulations is relevant to the imbibition of brine into a CO_2 saturated rock. This is of major relevance to the trapping of CO_2 in carbon sequestration applications (Nordbotten and Celia, 2012; Juanes et al., 2006; Taku Ide et al., 2007). The second set of simulations considers properties relevant to waterflooding of an oil saturated rock, which is relevant to hydrocarbon recovery applications. Finally, the third set of simulations comprises a sensitivity analysis where we analyse how changes in the viscosity ratio and capillary pressure curves impact the accuracy of the hybrid model.

4.4.1 *Description of the Numerical Experiments*

Numerical simulations were carried out in MRST. We simulate incompressible two-phase flow in a one-dimensional and uniform domain. No-flow boundary conditions were set at the rightmost cell of the domain. Saturation was prescribed as $S_0 = 1 - S_{nr}$ at the leftmost cell, which represents a fracture fully saturated by a wetting phase. The remaining cells are found initially at residual wetting phase saturation. There is no forced flow imposed in the domain, i.e. this set of initial and boundary conditions leads to counter current spontaneous imbibition. The domain size was set as $L_x = L_c = 2.5\ m$ for all the simulations. In the x-direction, i.e. the direction in which the wetting phase propagates, 1000 grid blocks were used ($N_x = 1000$). We consider the usual Corey parametrizations for the capillary pressure and relative permeability curves (see Section 1.2). We note that these models are used to match experimental data. We consider the following sets of parameters to evaluate the results given by the imbibition models:

- CO_2-brine Set of Simulations

  Four CO_2-brine simulations are carried out using experimental data from Bennion and Bachu (2010) (Table 4.1). Viscosities are taken as $\mu_w = 0.342$ cP and $\mu_n = 0.07$ cP. Porosity and permeability are fixed at $\phi = 0.15$ and $k = 100$ mD, respectively. Capillary pressure and relative permeability curves were fitted against experimental data for the imbibition of brine into CO_2 saturated carbonate rocks using Equations (1.5). The selected cases show different shapes of diffusion coefficient curves. We label the CO_2-brine simulations as $BC1$ to $BC4$.

- Waterflooding Set of Simulations

  Although this thesis is focused on CO_2 storage, we also evaluate here the performance of the imbibition model with parameters relevant to hydrocarbon recovery applications. Five waterflooding simulations are carried out using experimental data from Dernaika et al. (2013). Viscosities are taken as $\mu_w = 1$ cP and $\mu_n = 2.3824$ cP, which are respectively reference values for pure water and crude Brent oil (kinematic viscosity of $\nu = 2.86\ mm^2/s$). The permeability is fixed at $k = 100\ mD$. The porosity changes from simulation to simulation. Capillary pressure and relative permeability curves were fitted against experimental data for waterflooding in oil saturated carbonate rocks using Equation (1.5). We label the waterflooding simulations as $WO1$ to $WO5$ (Table 4.2).

- Sensitivity Analysis Set of Simulations

For the sensitivity analysis, the viscosity ratio is changed by varying the non-wetting phase viscosity while keeping wetting phase constant. The pore-size distribution index n_p (see Equation (1.5)) is also changed within a set of three values. The cases are labeled as LnVm where n is a natural number ranging from 1 to 3 and m is a natural number ranging from 1 to 5. Each value corresponds to a different value of viscosity ratio and power-law index. Tables 4.3 and 4.4 show the values of these parameters.

Case	Data in Bennion and Bachu (2010)	S_{wr}	S_{nr}	k_{rw}^{max}	k_{rn}^{max}	n_w	n_n	P_e [kPa]	n_p
$BC1$	Wabamum #3	0.825	0.045	0.9165	0.1	6.25	1.72	35	0.5
$BC2$	Winnipegosis	0.2	0.4149	0.1346	0.6	1.65	1.55	55	0.5
$BC3$	Grosmont	0.53	0.356	0.0249	0.1	2.25	2.76	14.5	0.5
$BC4$	Cooking-Lake #2	0.6	0.268	0.0788	0.1	1.7	1.15	69	0.5

Table 4.1: Parameters representative for CO_2 storage in saline aquifers.

Case	Data in Dernaika et al. (2013)	ϕ	S_{wr}	S_{nr}	k_{rw}^{max}	k_{rn}^{max}	n_w	n_n	P_e [kPa]	n_p
$WO1$	RT1-113	0.174	0.075	0.15	0.28	0.57	2.1	2.7	20	0.5
$WO2$	RT2-9	0.279	0.05	0.08	0.35	0.725	3	4	60	0.7
$WO3$	RT3-22	0.212	0.21	0.05	0.425	1	5	5	100	0.7
$WO4$	RT4-4	0.227	0.175	0.12	0.32	0.75	5	4	100	0.7
$WO4$	RT5-138	0.184	0.09	0.2	0.2	0.81	5	3	175	0.5

Table 4.2: Parameters representative for petroleum production through water flooding.

ϕ	μ_w [cP]	k [mD]	S_{wr}	S_{nr}	k_{rw}^{max}	k_{rn}^{max}	n_w	n_n	P_e [kPa]
0.1	1	100	0	0	1	1	2	2	100

Table 4.3: General parameters for the sensitivity analysis.

Case	μ_n [cP]	μ_w/μ_n	n_p
L1V1	100	0.01	0.1
L1V2	10	0.1	0.1
L1V3	1	1	0.1
L1V4	0.1	10	0.1
L1V5	0.01	100	0.1
L2V1	100	0.01	0.5
L2V2	10	0.1	0.5
L2V3	1	1	0.5
L2V4	0.1	10	0.5
L2V5	0.01	100	0.5
L3V1	100	0.01	0.9
L3V2	10	0.1	0.9
L3V3	1	1	0.9
L3V4	0.1	10	0.9
L3V5	0.01	100	0.9

Table 4.4: Viscosity ratio and capillary power-law index values for the sensitivity analysis.

4.4.2 *Measures for the Shape of the Diffusion Coefficient Curve*

Figure 4.7 shows the diffusion coefficient curves for the simulation cases specified in the previous sections. The diffusion coefficient curves clearly span a wide range of shapes. To obtain a parametrization we define two measures for the shapes of the diffusion coefficient curves, μ and σ. The parameter μ, given by $\mu = \frac{S_w^{max} - S_{wr}}{1 - S_{nr} - S_{wr}}$ is a measure of the position of the maximum within the saturation range. The parameter σ, given by $\sigma = \frac{\int_{S_{wr}}^{1-S_{nr}} \Pi(S_w) dS_w}{1 - S_{nr} - S_{wr}}$ with $\Pi(S_w) = 1$ if $D_w(S_w) \geq 0.1 \cdot max(D_w(S_w))$ and $\Pi(S_w) = 0$ otherwise, is a measure of the "flatness" of the curve. μ and σ are real numbers in the interval $[0, 1]$ and are sufficient to characterize the diffusion coefficient curves (see Figure 4.8).

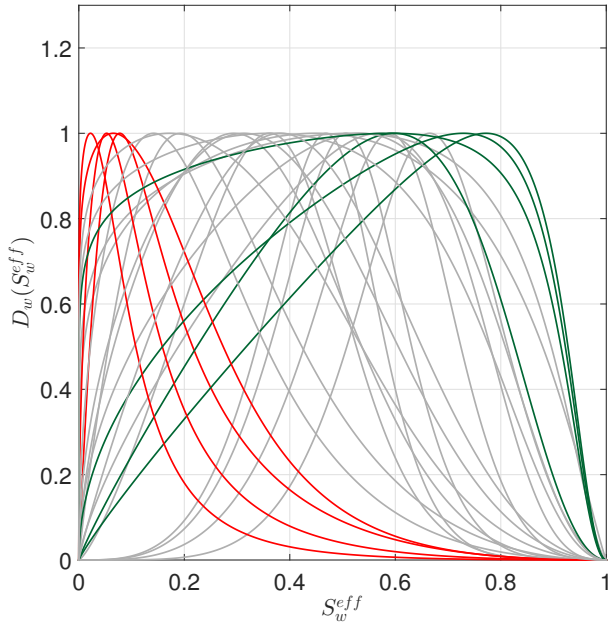

Figure 4.7: Illustration of the normalised diffusion coefficients for parameters from Tables 4.1, 4.2, 4.3 and 4.4. Dark green curves correspond to cases with high μ and σ while the red ones correspond to cases with low values of μ and σ.

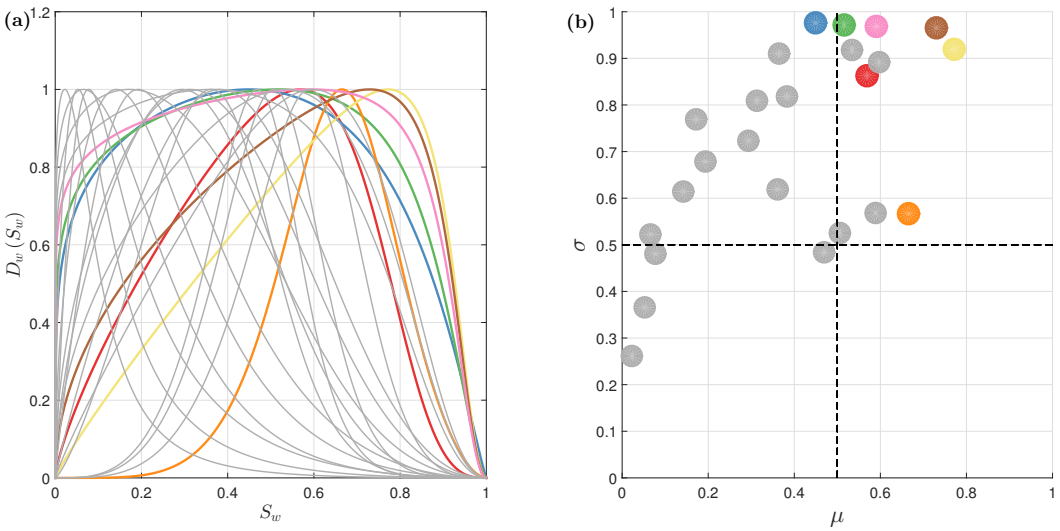

Figure 4.8: Illustration of the normalised diffusion coefficient functions (a) for parameters from Tables 4.1, 4.2, 4.3 and 4.4 and their characterisation to the position of the maximum and the width (b). Selected cases are highlighted in both plots through identical colours.

4.4.3 *Results*

The effectiveness of the filling back procedure and the improvement of the new imbibition function over a standard exponential function are discussed in this section. We first analyse the extreme cases by selecting the four diffusion curves with highest values for μ and σ and accordingly the other four cases with the

lowest values for these parameters (Figure 4.7). We calculate the errors incurred when estimating the transition time. We define the error in the estimate as $E \equiv |(\hat{t}_{true} - \hat{t}_{estimated}) / \hat{t}_{true}|$. We compare the estimated $\hat{t}$ with the true value given by high resolution numerical simulation. Figure 4.9 presents the relative errors for all the simulation cases. The maximum error in the estimation of $\hat{t}$ is 33%. A clear pattern is seen in the behaviour of the errors as it increases with smaller values for μ and σ (Figure 4.9(a)). The largest errors occur when the saturation front of the imbibed phase has a long tail at low saturations. For those cases, reconstructing the saturation front with a plateau leads to higher errors. As we fill the domain with wetting phase, the saturation increases globally in the physical domain since the overall level of wetting phase is low. The estimation provides better results if we consider piston-like shapes of the saturation front of the imbibed phase, which correspond to high values of μ and σ. Considering the physical parameters, better estimates (i.e. smaller errors) are related to high viscosity ratios and lower values of n_p (Figure 4.9(b)), since increasing the viscosity ratio and n_p shifts the diffusion coefficient curve to the right part of the saturation range.

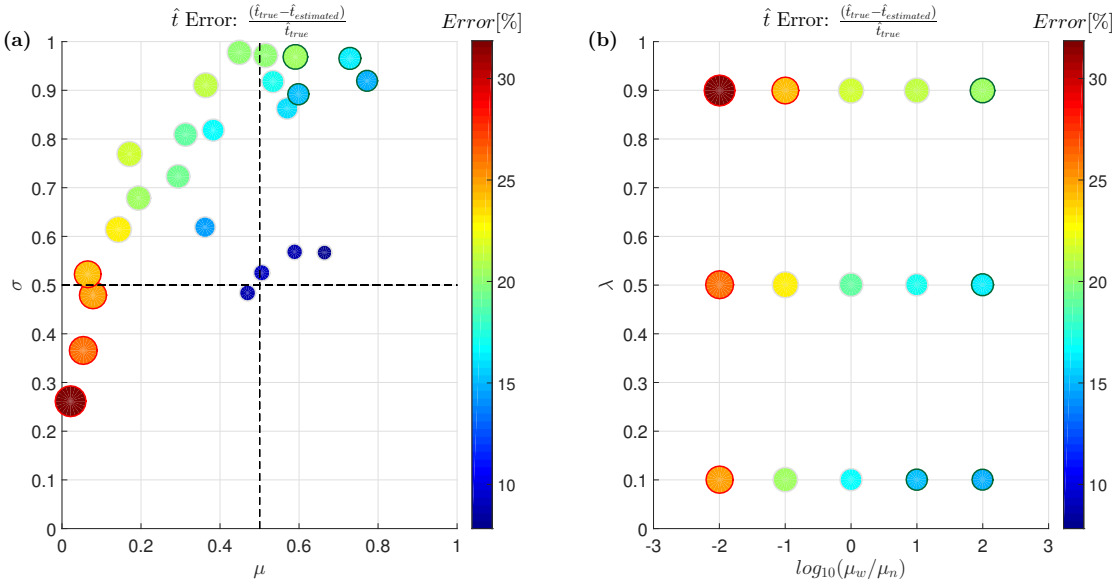

Figure 4.9: Relative error of the predicted $\hat{t}$ with respect to shape parameters (a) and physical parameters (b). The error is represented through colours and radii of the circles. Edges of the circles are coloured red and green to identify them with the coloured diffusion coefficient curves of Figure 4.7.

The improvement over an exponential model follows the same pattern (Figure 4.10). The improvement is a number between 0 and 1, with the first meaning no improvement over the exponential model and the latter meaning a perfect fit to the simulation. It is defined as $I \equiv |(E_{Aronofsky} - E_{hybrid}) / E_{Aronofsky}|$, where E is the integral over time of the difference between the imbibition curves of each method and the one provided by the simulation. Significant improvements,

with values close to 1 are observed mainly for the cases concentrated at the upper-right quarter of the $\mu - \sigma$ map, i.e. for simulations with high viscosity ratios. For these cases, displacement is piston-like. This leads to faster invasion of the wetting phase and most of the imbibition time is dominated by diffusion. Since we are matching the wetting phase front exactly at early time, very good improvements are observed in these cases.

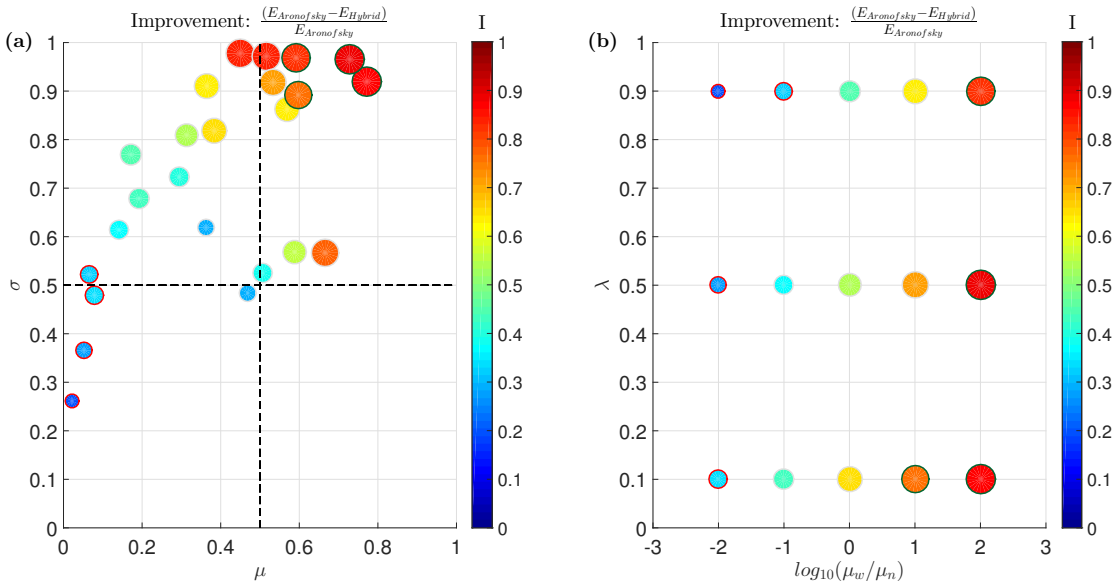

Figure 4.10: Improvement of the representation of the imbibition process of the hybrid model over the sigmoidal model with respect to shape parameters (a) and physical parameters (b). The improvement is represented through colours and radii of the circles. Edges of the circles are coloured red and green to identify them with the coloured diffusion coefficient curves of Figure 4.7.

We now investigate the relative imbibed volumes (Figure 4.11) for the eight selected cases shown in Figure 4.7. The best agreements with the simulation results are associated with large periods of diffusive displacement and the transition to late time occurs when more than 60% of the non-wetting phase has been already recovered. Furthermore, for those cases the sigmoidal model provides a fairly good approximation for the late-time behaviour, which leads to good overall matches between the predicted and true imbibition curves. On the other hand, the cases with worse agreement have a longer overall imbibition time, so the benefit of matching the early-time behaviour decreases. For those cases, it is clear that the late-time behaviour is not well represented by an exponential model. Despite providing a reasonable general scaling for the process, the late-time model given by Equation (4.16) does not provide an accurate representation of imbibition in the later stages of the process. A significant improvement with refined fitting procedures for the parameter γ is impossible as the procedures do not address the wrong representation of the general scaling. We point out, though, that the exponential model is generally used to model the entire range of imbibition (see, e.g. Kazemi et al. (1992)). Hence the improvement that our proposed model

offers is particularly noteworthy. Also, considering that in these cases the early time still corresponds to very large physical time scales, the good match at early time may still be sufficient for practical purposes such as improved oil recovery simulations. It is also noteworthy that the filling back procedure provides fairly good estimates of the transition time for all cases.

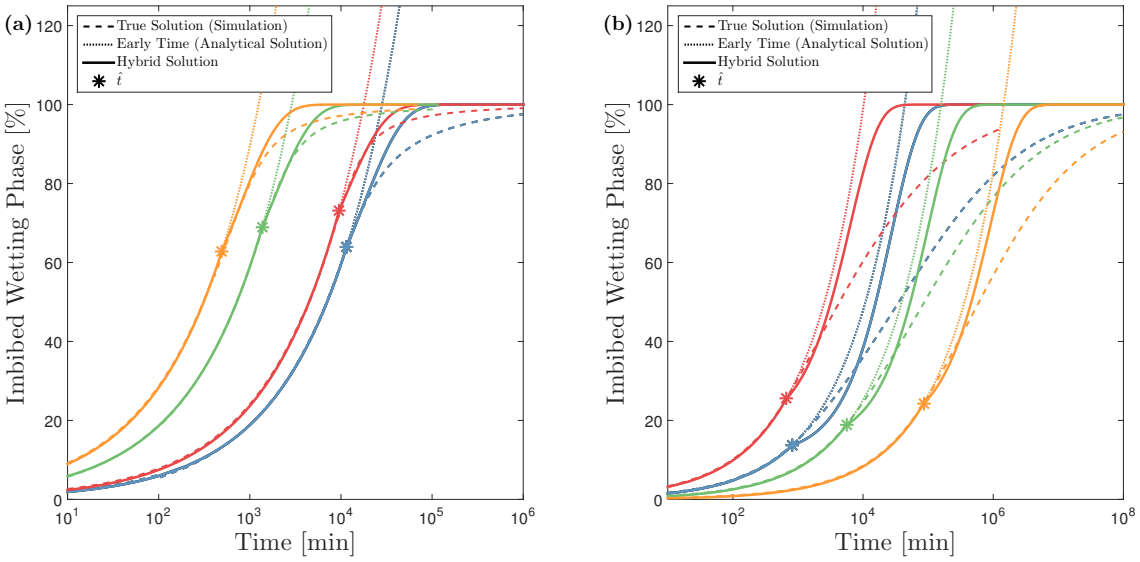

Figure 4.11: Representative curves for the imbibed volume fraction for high values of μ and σ (a) and low values of μ and σ (b).

The hybrid imbibition model shown in this work provides better results for right-skewed flat diffusion coefficient rates. Since increasing the wetting/non-wetting phase viscosity ratio shifts the diffusion coefficient curve to the right part of the saturation range, the hybrid model is particularly valid when the viscosity ratio is high. This scenario would include applications from the imbibition of brine into a CO_2 saturated rock, to several hydrocarbon recovery scenarios where the non-wetting phase has a lower viscosity than the wetting phase, and even the study of the imbibition of fracturing fluids into low permeability rocks.

4.5 SUMMARY AND CONCLUSIONS

One of the key mechanisms that govern matrix-fracture fluid exchange after CO_2 injection stops is counter-current spontaneous imbibition, caused by the rise in the brine level as the CO_2 plume travels upwards to the cap-rock. In this chapter, a hybrid model that captures both early- and late-time regimes of fluid transfer due to counter-current spontaneous imbibition was presented. Early-time imbibition is governed by capillary diffusion of the wetting phase into a semi-infinite domain. The self-similarity of this process is used to obtain a semi-analytical solution. The late time behaviour of the process is approximated by an exponential model.

The hybrid model switches from early-time to late-time behavior at the transition time $\hat{t}$ when the imbibed volume deviates from the $t^{1/2}$ scaling. A procedure to determine this transition time based on the semi-analytical solution of the early-time regime has been developed. The procedure obtains the transition time from the moment when the saturation $\hat{S}_w$ corresponding to 60% of the cumulative diffusion has reached the boundary. In order to approximate this moment from the semi-analytical solution we have introduced the Filling Back Procedure.

A number of cases arising from different applications such as hydrocarbon recovery and CO_2-brine flow were considered to evaluate the accuracy of the newly developed model. Results show that the filling back procedure is effective in the prediction of the true value, providing errors that are always less than 33%. This is acceptable for our application, since even though we may be underestimating the transition time, we still benefit from the perfect matching at early time to provide an imbibition model that is at least as good as the widely used exponential approximation. Improvement over estimates based on the sigmoidal function is remarkably high for cases with high viscosity ratios, which makes the hybrid model especially favourable for modelling brine imbibition in CO_2 saturated rocks during CO_2 storage simulations.

The implementation of the hybrid recovery function in dual-porosity simulators is possible but not addressed in this thesis. A derivation of a transfer function would involve differentiating the imbibition model given by Equation 4.17 with respect to time. Note that since Equation 4.17 has terms that do not depend on time via an exponential function, such as $H(t)$ and $E(t)$, the final transfer function would also have an explicit dependence on time; in contrast to typical first-order models that depend only on saturation or pressure in the matrix. This is a weakness of this model, since it leads to challenges in the computational implementation. A possible way to track the elapsed time since imbibition started is to have a field defined for each cell that counts the time elapsed since the wetting phase first arrived at the cell. Nonetheless, while not necessarily straightforward, the implementation of the hybrid model presented in this chapter in a dual-porosity simulator would provide more accurate descriptions of the saturation fields in the fracture system and matrix, hence leading to a better engineering design.

5

CO_2 STORAGE POTENTIAL IN NATURALLY FRACTURED RESERVOIRS

5.1 INTRODUCTION

The previous chapters presented novel transfer functions that capture accurately the transfer of CO_2 and brine due to gravity drainage and spontaneous imbibition. During injection of CO_2 in NFRs, the utilization of such a transfer function is crucial to the correct evaluation of CO_2 storage in NFRs. In Chapter 3 it was shown on a matrix block scale that currently existing formulations for transfer due to gravity drainage fail to reproduce the results given by simulations of fully-resolved blocks. It was also shown that this misrepresentation of the transfer leads to significantly different results on the field scale. Finally, Chapter 3 presented the results of CO_2 injection in a conceptual fractured anticline model. It was illustrated with numerical simulations that the transfer rate coefficient β has a significant impact on the amount of CO_2 that spills through the boundaries of the anticline, and therefore, the modeling of this parameter is a key ingredient necessary for the evaluation of storage projects in NFRs. These results corroborate the results for the Johansen formation shown in Chapter 1.

In this chapter we present a more comprehensive analysis of storage in large fractured aquifers. This analysis is based on the work of Szulczewski et al. (2012), where the CO_2 storage potential in the U.S. was evaluated by considering a set of large aquifers that were identified in sedimentary basins across the country. While there is no evidence of fractures in these aquifers, they represent a vast set of aquifers with different properties and are located in different geological environments that can be used to understand the impact of fractures in field-scale storage of CO_2.

The main objective of this chapter is to evaluate the impact of a fracture network in the target formation on the storage capacity. In Section 5.2, in order to simulate the plume migration in the fractures and in the matrix, we present the development of a novel vertically-integrated dual-porosity model (VIDP). This model uses the transfer function developed in Chapter 3 to evaluate the storage in the fractured aquifers without running the full field-scale simulation.

Vertically-integrated models reduce the dimension of the system by assuming vertical equilibrium in the aquifer. Vertical equilibrium may often be assumed in fractures due to its high permeability, that leads to fast segregation of phases (Nordbotten and Celia, 2012). Here, as in Szulczewski et al. (2012), the vertical equilibrium assumption reduces significantly the complexity of the numerical computations while preserving the main features of the flow. In Section 5.3 we evaluate the storage in hypothetical fractured aquifers located in different geological environments with different petrophysical properties. The results shown in this section allow the understanding of the best geological conditions for CO_2 storage in NFRs. Finally, in Section 5.4 we compare CO_2 storage in fractured and unfractured aquifers, using real aquifers documented by Szulczewski et al. (2012). The results shown in this section allow the evaluation of the impact of fractures in CO_2 storage operations and provide an assessment about the feasibility of CO_2 storage in NFRs.

5.2 A VERTICALLY-INTEGRATED MODEL FOR CO2 STORAGE IN DUAL-POROSITY RESERVOIRS (VIDP)

The solution of the multiphase system defined by Equations (1.3) is necessary to understand the CO_2 flow dynamics in NFRs. In many cases, however, solving the full coupled system may be computationally expensive; particularly for large aquifers. Under some assumptions, it is possible to make use of simple semi-analytical solutions that reduce the complexity of this system. This helps to gain insight about the first-order physics behind the migration of the CO_2 plume in large aquifers

A widely used family of such methods are the vertically-integrated models (Nordbotten and Celia, 2012; Guo et al., 2014; Szulczewski et al., 2012; Juanes and MacMinn, 2008). Such models make specific assumptions about the distribution of brine and CO_2 across the vertical cross section. This allows for an integration over the vertical dimension and reduces the dimensionality of the problem by one. The dimensionality reduction allows a faster computation of the dynamics of the system, while preserving the key features of the distribution of the phases.

The reconstruction of the saturation profile in the vertical direction requires simplifying assumptions about the vertical dynamics and equilibrium of phases. If the vertical dynamics happens significantly faster than the plume migration in the horizontal plane, a reasonable assumption is that phases are in equilibrium, that is, that the pressure distribution for each fluid is hydrostatic. This is known

as the *Dupuit* approximation. The segregation timescale may be estimated as (Guo et al., 2014)

$$t_s \approx \frac{L_z \phi \mu_w}{k_{rw} k_z \Delta\rho g}, \tag{5.1}$$

where L_z is the aquifer thickness, ϕ the average porosity of the medium, μ_w the average viscosity of brine, k_{rw} the endpoint of the relative permeability of the formation to the brine phase, k_z the vertical permeability of the aquifer, $\Delta\rho$ the density difference between brine and CO_2 phases and g the magnitude of gravity acceleration. For some aquifers, the vertical equilibrium assumption may be inappropriate, and hence other techniques to include vertical dynamics are necessary (Guo et al., 2014). Physical assumptions relative to the equilibrium distribution of phases are also necessary in order to define a vertically-integrated model. Regarding the distribution of fluids in equilibrium, if capillary pressure in the medium is low, the sharp-interface approximation may be considered (Nordbotten and Celia, 2012). The sharp interface assumption considers that there is no transition zone (capillary fringe) between brine and CO_2 phases, that is, the medium is saturated at each point either by brine or by CO_2 with residual brine. This approximation may be suitable if the capillary pressure in the aquifer is low.

Within the fracture system, both vertical equilibrium and sharp interface assumptions are reasonable. Fractures will generally have high permeability and low porosity, which shortens the segregation time. An estimation of t_s in the fractures can be made in order to support this statement. Realistic (and very conservative) values for fractured aquifers and fluids are: $L_z = 1000$ m, $\phi = \phi_f = 0.01$, $k_z = k_f = 400$ mD, $\Delta\rho = 262$ kg/m^3, $k_{rw} = 1$ and $\mu_w = 0.644$ cP. Porosity and permeability of the fracture system were taken from data that was estimated for the In-Salah CO_2 storage project via discrete fracture network and history matching (Iding and Ringrose, 2010; Pamukcu et al., 2011). The density difference and brine viscosity correspond to representative values for deep formations in cold basins, and high-salinity brines (Nordbotten and Celia, 2012). These values lead to a segregation time of $t_s \approx 0.2$ year, which is substantially lower than the injection times, typically of the order of tens of years. Fractures also typically have capillary pressures that are many orders of magnitude lower than in the matrix and that are neglected in most studies (Rossen and Kumar, 1992). Therefore, no transition zone between CO_2 and brine is likely to be found within the fractures.

Considering the assumptions of vertical equilibrium and sharp interface, the one-dimensional vertically-integrated mass balance equation for the CO_2 phase

during the CO_2 injection period for an unfractured formation is given by (Szulczewski et al., 2012)

$$\phi\left(1-S_{wr}\right)\frac{\partial h_n}{\partial t}+\frac{\partial}{\partial x}\left(f_n q_{inj}-k\Delta\rho g L_z\bar{\lambda_n}\left(1-f_n\right)\frac{\partial h_n}{\partial x}\right)=0, \tag{5.2}$$

where $h_n = h_n\left(x,t\right)$ is the thickness of the CO_2 plume and q_{inj} $[L^2/t]$ is the integrated injection rate. The injection rate is a given input parameter of the CO_2 plume model, and is determined for each case by considering pressure and migration limitations, as shown in the next sections. The vertically-integrated fractional flow function is defined as $f_n = \bar{\lambda_n}/(\bar{\lambda_w}+\bar{\lambda_n})$, where the average mobilities are defined as

$$\bar{\lambda_n}=\frac{k_{rn}^{max}}{\mu_n}\frac{h_n}{L_z},\ \bar{\lambda_w}=\frac{1}{\mu_w}\frac{L_z-h_n}{L_z}. \tag{5.3}$$

The diffusive term in Equation (5.2) may be neglected when the mobility ratio $M = (1/\mu_w)/(k_{rn}^{max}/\mu_n) \ll 1$ (Juanes and MacMinn, 2008). Therefore, Equation (5.2) reduces to a nonlinear advection equation given by

$$\phi\left(1-S_{wr}\right)\frac{\partial h_n}{\partial t}+\frac{\partial}{\partial x}\left(f_n q_{inj}\right)=0. \tag{5.4}$$

Equation (5.4) models the CO_2 plume migration in a single continuum. The extension of this model for fractured reservoirs involves writing a second equation that represents the accumulation of CO_2 in the matrix. The vertically-integrated dual-porosity model (VIDP) developed in this thesis is given by

$$\phi_f\frac{\partial h_{nf}}{\partial t}+\frac{\partial}{\partial x}\left(f_n q_{inj}\right)=-\overline{T}_{nm}, \tag{5.5}$$

$$\phi_m\left(1-S_{wr}\right)\frac{\partial h_{nm}}{\partial t}=+\overline{T}_{nm}, \tag{5.6}$$

where h_{nf} and h_{nm} represent the thickness of the CO_2 plume in the fracture and matrix, respectively, and $\overline{T}_{nm}$ [L/t] is the vertically-integrated matrix-fracture transfer rate. The transfer rate is modeled as

$$\overline{T}_{nm}=\phi_m\left(1-S_{wr}\right)\beta\left(h_{nm}^{max}-h_{nm}\right), \tag{5.7}$$

where h_{nm}^{max} is the maximum penetration depth of CO_2 in the matrix and β is the drainage rate coefficient that is calculated via the model developed in Chapter 3, Equation (3.9). The maximum penetration depth h_{nm}^{max} is calculated as

$$h_{nm}^{max}=h_{nf}-\frac{P_e}{\Delta\rho g}, \tag{5.8}$$

where P_e is the capillary entry pressure. Note that h_{nf} is equivalent to H from Chapter 3. At the left boundary $h_{nf} = L_z$ holds. Three particular assumptions of this model require discussion:

1. **The Vertically-integrated plume migration domain is one-dimensional.**

   Following the developments in Szulczewski et al. (2012), we consider large aquifers with horizontal dimensions much larger than the vertical dimension, which allows the application of vertically-integrated models (Nordbotten and Celia, 2012). Furthermore, we assume that the system may be further simplified by considering symmetry in the y-direction. This may be achieved if CO_2 is injected through a line-driven array of wells; a sequence of a large number of wells with small spacing between them. The small spacing would lead to fast equilibration of the CO_2 between the wells, and the final plume would flow in an approximately one-dimensional displacement in the x-direction (the direction perpendicular to the array of wells). This hypothetical scenario was considered in Szulczewski et al. (2012) to consider extremely large storage projects that could decarbonize the United States. It is therefore suitable for country-wide storage estimates. If only one well is to be considered, Equation 5.5 should be rewritten in a cylindrical coordinate system.

2. **Capillary continuity across horizontal fractures.**

   It is assumed in this model that horizontal fractures have apertures that are small enough to establish capillary bridges across the vertical blocks. Therefore, a vertical stack of blocks behaves as a single tall block (often called a match stick) spanning the entire thickness of the aquifer. This means that the effective buoyancy column that drives CO_2 to drain the matrix is a function of the density difference and the thickness of the plume in the fractures. The impact of horizontal fractures on the rate of the drainage process is not taken into account here. However, we acknowledge that even horizontal fractures that allow for capillary bridges may retard the drainage by reducing the permeability between the blocks. In Labastie (1990), experiments show that capillary continuity occurs for a fracture aperture less than 0.3 mm for a surface tension of 24 mN/m. The empirical model developed in Barton et al. (1985) for an initially unloaded mechanical aperture E_0 allows to estimate the order of magnitude of fracture aperture. The model determines E_0 by

$$E_0 \approx \frac{JRC}{5}(0.2\frac{\sigma_c}{JCS} - 0.1), \tag{5.9}$$

where E_0 is the aperture in millimeters, JRC is a dimensionless parameter ranging typically from 0 to 15 and depending on the roughness of the fractured walls, σ_c is the unconfined compression strength and JCS is the joint wall compression strength. We assume for simplicity unweathered joints, which gives $JCS = \sigma_c$. The model then reduces to $E_0 \approx 0.02 \cdot JRC$. Therefore, according to this model, the initial unloaded aperture takes its maximum at $JRC \approx 15$, which gives $E_0 \approx 0.3$ mm, which is the threshold value measured in Labastie (1990). In-situ fracture apertures will generally be smaller than this value due to compressive stress caused by the overburden. While further experimental research taking into consideration conditions relevant to CO_2 storage operations are needed to evaluate the establishment of capillary bridges, the phenomenon is established in general (Aspenes et al., 2008) and is likely to occur also with supercritical CO_2 and brine.

3. **Sharp interface inside the matrix.**

   It is assumed here that the sharp interface approximation also holds for the matrix. This means that inside the matrix there is a sharp transition between a region with $S_{wm} = 1$ and a region with $S_{nm} = 1 - S_{wr}$. This assumption holds for monodisperse pore space. While this assumption is actually dependent on the petrophysical properties of the formation, the matrix will generally be drained to saturations close to $(1 - S_{wr})$ above h_{nm}^{max}. We acknowledge that this assumption does not lead to a conservative estimate. However, it does not change the qualitative results, nor the order of magnitude of the quantitative results.

Note that the implementation of dual-porosity models with match-stick like matrix blocks in conventional 3D/non-integrated simulators is not trivial. In conventional simulators, dual-porosity models generally assume some level of locality. In a discrete formulation, this implies that the transfer between matrix and fractures in a grid cell depends only on variables associated with this cell. Therefore, conventional dual-porosity models are typically not readily prepared to represent matrix blocks that are larger than the simulation grid blocks and span several cells. In the VIDP model, however, since the information of plume thickness is available for every cell in the horizontal dimension (h_{nf} is a function of x and t), match-stick blocks may be straightforwardly implemented. Hence, while VIDP represents a simplified model that is in principle less general than conventional simulators, it allows the simulation of flow in NFRs with long blocks spanning the whole aquifer thickness. This is an useful feature to estimate CO_2 storage in NFRs. Note also that the assumption of match-stick blocks implies that CO_2 segregates in the vertical fractures and flows around the blocks without penetrating in the horizontal fractures. This assumption also relies on

the small aperture of horizontal fractures caused by the overburden stress. Small apertures imply a higher capillary entry pressure in the horizontal fractures, making the vertical fractures a preferential pathway for CO_2 flow.

5.2.1 *Implementation of the VIDP model in the AD Framework of MRST*

The VIDP model is implemented in the automatic-differentiation framework of MRST. MRST provides a framework that allows the rapid prototyping and experimentation of models relevant to reservoir simulation (Krogstad et al., 2015).

A class called `PlumeSimulatorDP` was developed that receives in its constructor two structures: `aquifer` and `dp_info`. The `aquifer` structure contains the basic properties of an aquifer and the basin in which it is located; such as its depth, rock properties and the temperature of the basin. The `dp_info` structure contains two fields: `continuum1` and `continuum2`, the first representing properties (porosity and permeability) of the flowing continuum (fractures) and the second properties of the second continuum. The code is flexible and in case `dp_info` has only the `continuum1` field, it understands it as a single-porosity simulation. Apart from the constructor, the other methods available in the `PlumeSimulatorDP` are:

- `setupSimulation`: creates the function handles to evaluate the residual of the equations.
- `advancePlume`: advances a state of the CO_2 plume by one time step.
- `integrateTime`: makes successive calls to `advancePlume` in order to integrate the PDE's over a certain time range.

The full code of the `PlumeSimulatorDP` class as well as an example script for running a CO_2 plume migration simulation with this code are provided in Appendix C. Figure 5.1 shows the results that are produced by the example script, corresponding to the CO_2 profile in the matrix and in the fractures after 10 years of injection in a model aquifer.

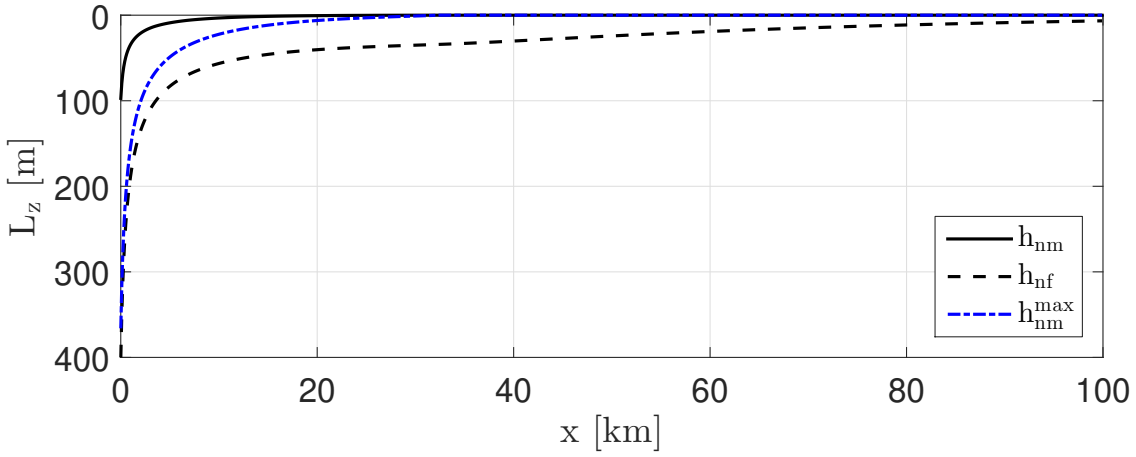

Figure 5.1: CO_2 plumes in the fracture and matrix after 10 years of injection. Output of VIDP simulation script available in Appendix C.

The upwind approximation is used to evaluate the nonlinear flux $f_n q_{inj}$ at the faces of each finite volume. An implicit Euler scheme is used to discretize the time operator. The automatic differentiation framework of MRST ensures that for every set of operations that is carried out, the derivative of the residual with respect to the primary variables is calculated. Therefore, the nonlinear Newton iterations are simply carried out by applying the nonlinear operators to the field variables and then retrieving the Jacobian. The following code illustrates the nonlinear loop to solve the nonlinear system and advance one timestep:

```
%% Nonlinear Newton iteration for advancing a timestep
while (resNorm > tol) && (nit < maxits)

    %% Maximum penetration depth in the matrix
    h2max = double(h1_ad)-Pe/(delta_rho*g);
    h2max(h2max<0) = 0;

    %% Equation 1 - fractures, Equation 2 - matrix
    eq1 = ps.equations.h1Eq(h1_ad,h10,h2_ad,h2max,dt,qinj);
    eq2 = ps.equations.h2Eq(h2_ad,h20,h1_ad,h2max,dt);

    %% Applying boundary condition - hnf = Lz in the left boundary
    bc1l = ps.equations.BCFluxLeft(h1_ad(1),h10(1),h2_ad(1),...
                                   h2min(1),h2max(1),dt,qinj);
    eq1(1) = eq1(1) - bc1l;

    %% Concatenating equations
    eqs = {eq1, eq2};
    eq  = cat(eqs{:});

    %% Solving system
```

```
    J   = eq.jac{1};  % Jacobian
    res = eq.val;     % residual
    upd = -(J \ res); % Newton update

    %% Updating variables
    h1_ad.val = h1_ad.val + upd(h1Ix);
    h2_ad.val = h2_ad.val + upd(h2Ix);

    %% Residual and number of iteration controls
    resNorm = norm(res);
    nit     = nit + 1;
end
```

The residual of equations `h1Eq`, `h2Eq`, as well as the transfer functions are outlined below:

```
%% Mobilities and vertically-integrated fractional flow function
mobg = @(hg)(krgmax/mug)*(hg/H);
mobw = @(hg)(1/muw)*(H-hg)/H;
fg = @(hg)(mobg(hg)./(mobg(hg)+mobw(hg)));
fluxf =  @(h,qinj) upw(qinj*fg(h),qinj>0);

%% Define transfer function handles
transferm = @(h2,h2max,beta)beta*(1-swc)*phi2*(h2max-h2);
transferf = @(h2,h2max,beta)transferm(h2,h2max,beta);

%% Equation for the fracture
ps.equations.h1Eq   = @(h1,h10,h2,h2min,h2max,dt,qinj)  (dx/dt)*phi1*(h1 - h
    10) ...
    + div(fluxf(h1,qinj)) + transferf(h2,h2max,beta);

%% Equation for the matrix
ps.equations.h2Eq   = @(h2,h20,h1,h2min,h2max,dt)  (dx/dt)*phi2*(1-swc)*(h2
    - h20) - ...
                            transferm(h2,h2max,beta);

%% Equation for the boundary condition
ps.equations.BCFluxLeft = @(h1l,h1l0,h2l,h2minl,h2maxl,dt,qinj) ...
                                    phi1*(h1l-h1l0)*dx+...
                                    qinj*dt*(fg(h1l)-fg(H)) + ...
                                   transferf(h2l,h2maxl,beta);
```

In the code, the discrete divergence and upwind operators are defined as in Krogstad et al. (2015).

5.2.2 *Validation*

The fact that there is no straightforward way to implement the match-stick dual-porosity transfer function in a conventional simulator limits the possibility for validating the VIDP. We hence compare the results given by the code with Juanes and MacMinn (2008) for an unfractured formation (single-porosity system).

The mobility ratio $M = \frac{1/\mu_w}{k_{rn}^{max}/\mu_n}$ is taken as $M = 0.1$ and a connate water saturation of $S_{wr} = 0.3$ is assumed. The results are shown in terms of dimensionless spatial variables in Figure 8 of Juanes and MacMinn (2008). Although the VIDP-AD implementation is dimensional, the results are divided by the spatial lengths L_x and L_z for comparison. Figure 5.2 shows the comparative results between the results shown in Juanes and MacMinn (2008) and the VIDP-AD implementation. An excellent agreement is seen, demonstrating that the newly developed code is capable of reproducing results available in the literature for simulation of vertically-integrated systems. Due to its novelty, there is no means of validating the dual-porosity behaviour of the VIDP-AD code. However, the results provided by the code have always shown the expected qualitative behaviour during the developments outlined in this chapter.

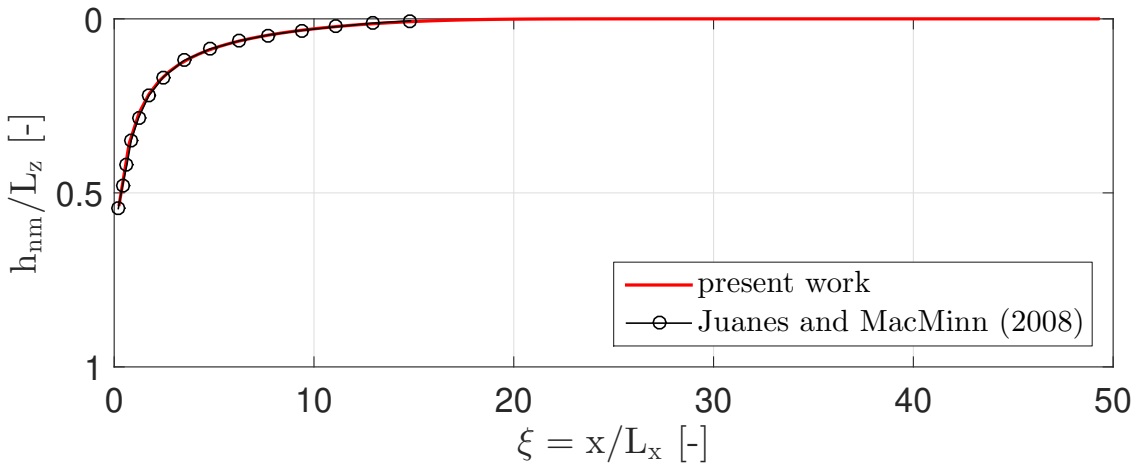

Figure 5.2: Validation of the VIDP-AD code. Comparison between results given by the code developed in this thesis and results presented in Figure 8 of Juanes and MacMinn (2008). Very good agreement is seen between the extent of the CO_2 plumes.

5.3 THE BEHAVIOUR OF CO2 STORAGE IN NFRS

In this section we investigate which environments are more favourable for CO_2 storage in a fractured reservoir. The density of CO_2 increases with pressure (and hence depth, if we consider normal subsurface pressures) and decreases with increasing temperature. Here, similar to Chapter 3, we define "warm" and "cold" basins based on the surface temperature T_s and geothermal gradient G_T. Warm

basins have $T_s = 20\ ^oC$ and $G_T = 45\ ^oC/km$. Cold basins have $T_s = 10\ ^oC$ and $G_T = 25\ ^oC/km$. We also consider aquifers that are at "shallow" depths ($D = 1000$ m) and "deep" depths ($D = 3500$ m) (see Figure 5.4). Figure 5.3 shows the CO_2 density profiles as a function of depth for warm and cold basins. CO_2 density is higher if the aquifer is deep and is situated on a cold basin. For an unfractured formation, this effect is advantageous, as a higher CO_2 density means a larger stored mass per unit volume. In a fractured formation, however, a higher CO_2 density also means a smaller density difference and hence a smaller buoyancy force that drives CO_2 into the matrix. This dual behaviour was briefly discussed in Chapter 3 for storage in a single matrix block. In this section it is investigated more thoroughly via numerical simulations with the VIDP-AD code.

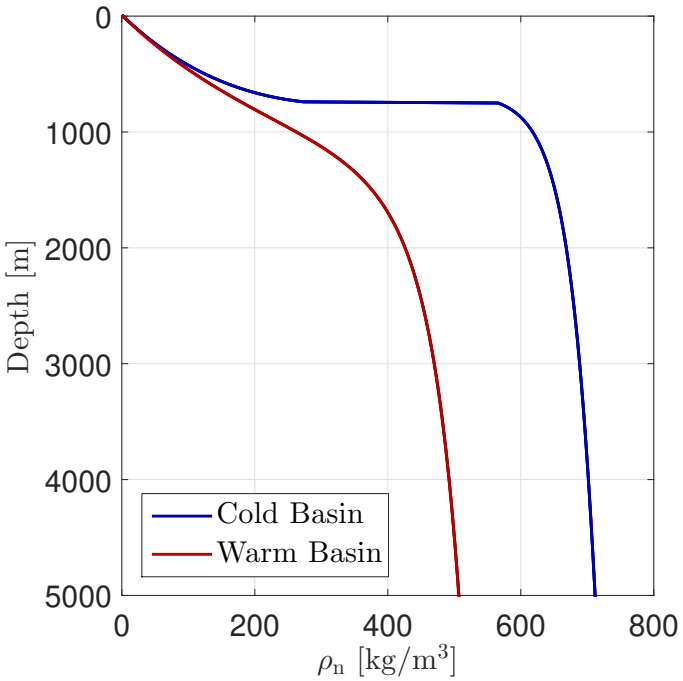

Figure 5.3: CO_2 density profiles as a function of depth for warm (red) and cold (blue) basins. Warm basins have a surface temperature of $T_s = 20\ ^oC$ and geothermal gradient of $G_T = 45\ ^oC/km$. Cold basins have a surface temperature of $T_s = 10\ ^oC$ and geothermal gradient of $G_T = 25\ ^oC/km$. The CO_2 density is calculated according to Nordbotten and Celia (2012).

The capillary entry pressure P_e is case-dependent and fixed for each simulation at "low", "medium" or "high" values: $P_e = 0$ kPa, $P_e = 75$ kPa and $P_e = 150$ kPa respectively. The transfer-rate coefficient β is also case-dependent and ranges from $\beta = 10^{-10}$ s^{-1} to $\beta = 10^{-6}$ s^{-1} by factors of 10. The average pressure of the aquifer is fixed at the hydrostatic pressure of brine, while the average temperature is defined by the temperature of the basin considering a constant thermal gradient and surface temperature. Density of the CO_2 phase is held constant for each simulation case and calculated as a function of the aquifer pressure and temperature: $\rho_n = \rho_n\left(P_{res}, T_{res}\right)$ according to the model of Spycher et al. (2003).

The other aquifer properties and operation parameters are fixed for all simulation cases and can be found in Table 5.1. CO_2 is injected at a constant rate during a period of T_{inj} = 10 years. The model was discretized with uniform grid cells of Δx = 30 m and a constant time step of Δt = 0.05 year. This discretization was shown to be sufficient to capture the dynamics for all the simulation cases.

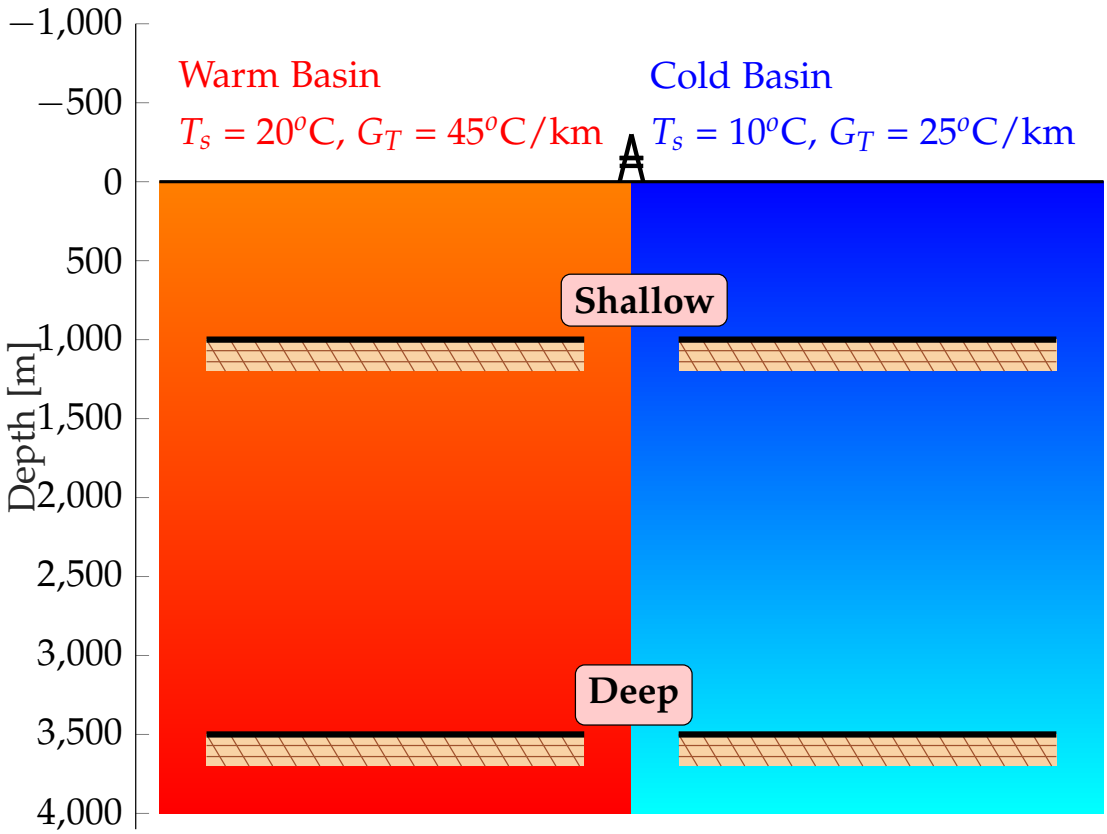

Figure 5.4: Selected fracture aquifers for analysis of CO_2 storage. We consider shallow (depth = 1000 m) and deep (depth = 3500 m) aquifers that are located in warm and cold basins. For each of the four environments, we consider three capillary pressure levels: P_e = 0 kPa, P_e = 75 kPa and P_e = 150 kPa. The transfer-rate coefficient β ranges from 10^{-10} s^{-1} to 10^{-6} s^{-1}

Parameter	Description/Source [units]	Value
L_x	Aquifer's horizontal extent [km]	300
L_z	Aquifer thickness [m]	50
W	Aquifer width [km]	100
ϕ_f	Fracture porosity [$-$] . Value based in history matched values (Pamukcu et al., 2011) and the SPE dual-porosity benchmark problem (Firoozabadi and Thomas, 1990).	0.01
ϕ_m	Matrix porosity [$-$]	0.2
μ_w	Brine viscosity [cP]	0.8
μ_n	CO_2 viscosity [cP]	0.06
k_{rn}^{max}	End-point of relative permeability to CO_2 phase [$-$].	0.6
S_{wr}	Brine residual saturation [$-$]	0.4
ρ_w	Brine density [kg/m^3]	1000
q_{inj}	Vertically-integrated injection rate [m^2/s]. As in Szulczewski et al. (2012), $q_{inj} = Q_{inj}/(2W)$, where Q_{inj} is the total injection rate of the array of wells.	0.00015
β	Transfer rate coefficient [$1/s$] (case-dependent). See Equation 5.7.	10^{-6}, 10^{-7}, 10^{-8}, 10^{-9} or 10^{-10}
P_e	Capillary entry pressure (case-dependent).	0, 75 or 150

Table 5.1: Parameters used in the simulations carried out in this section.

We first analyse the behaviour of the CO_2 plume in the fractured formation at different times during the CO_2 injection period and for different levels of capillary entry pressure and transfer rate coefficients. Figure 5.5 shows the temporal evolution of CO_2 plumes in the fractures and matrix for a formation with P_e = 75 kPa and $\beta = 10^{-6}$ s^{-1} at a shallow depth in a warm basin. CO_2 is injected at the left boundary into the fractures. It forms a plume in the fractures that provides enough buoyancy to overcome capillary entry pressure. As CO_2 transfers to the matrix, the water level rises. This creates the inflection point in the CO_2 plume seen at $x \approx 15$ m after one year of injection. The inflection point gives rise to a CO_2 "plateau" in the fractures; a region of constant CO_2 plume thickness in the fractures. The CO_2 plume is transported across the aquifer close to the cap rock, and reaches the boundary of the aquifer after ≈ 5 years. An interesting observation is that the CO_2 plume thickness in the fractures close to the well does not change significantly with time after the plume reaches the boundary. Note that close to the well the CO_2 plume has very similar profiles after $t = 5$ and $t = 10$ years of injection between $x = 0$ km and $x \approx 60$ km. This means that injected CO_2 flows close to the cap rock instead of contributing to the thickening of the plume close to the well. This reduces the CO_2 transfer rate to the matrix and overall storage in the aquifer, and most of the injected CO_2 shall escape through the boundary after the breakthrough of the CO_2 plume. In order to understand the shape of the CO_2 plume in the fracture system, we analyse the sensitivity of the plumes to different values of P_e and β in Figures 5.6 and 5.7.

Figure 5.6 shows the CO_2 plumes in the fractures at the end of injection time, T_i = 10 years, for different capillary entry pressures and a fixed β of 10^{-6} s^{-1}. Figure 5.7 shows the CO_2 plumes in the fractures at the end of injection time for different values of β and fixed P_e of 75 kPa. The CO_2 plume in the fractures is thinner at each point in the spatial direction when more CO_2 is transferred to the matrix. It is not surprising that higher capillary entry pressures lead to a smaller amount of CO_2 in the matrix (In Figure 5.6, the matrix contains more CO_2 for the P_e = 0 kPa case). The position of the CO_2 plume tip is also strongly affected by the maximum amount of CO_2 that the matrix is capable to store. In Figure 5.7, we observe that the thickness of the CO_2 plume in the fractures is strongly affected by the transfer-rate coefficient. As CO_2 transfers to the matrix, the plume gets thinner in the fractures; compare the plumes in the fractures for $\beta = 10^{-10}$ s^{-1} and $\beta = 10^{-7}$ s^{-1}, for instance. This even leads to a locally non-monotonic behaviour of the amount of CO_2 in the matrix with respect to β. While intuitively one could argue that higher CO_2 transfer rates will always lead to higher CO_2 in the matrix, we observe that after $\approx$ 10 km, there is more CO_2 in the matrix for the case with $\beta = 10^{-8}$ s^{-1} than for the case with $\beta = 10^{-7}$ s^{-1} due to a thicker plume in the fractures. In figures 5.6 and 5.7 we also show $L^{min}_{CO_2}$ (dotted lines), the minimum required level of CO_2 in the fractures for transfer (defined in Equation (5.10)). After the plume in the fractures reaches this thickness before the inflection point, transfer ceases and the plumes propagate with this thickness throughout the domain. There is in fact an intricate relationship between CO_2 storage in the NFRs and parameters such as the transfer-rate coefficient and the CO_2 plume in the fractures. This reinforces the value of developing quick tools for estimating CO_2 in NFRs with dual-porosity behaviour, such as the VIDP-AD code.

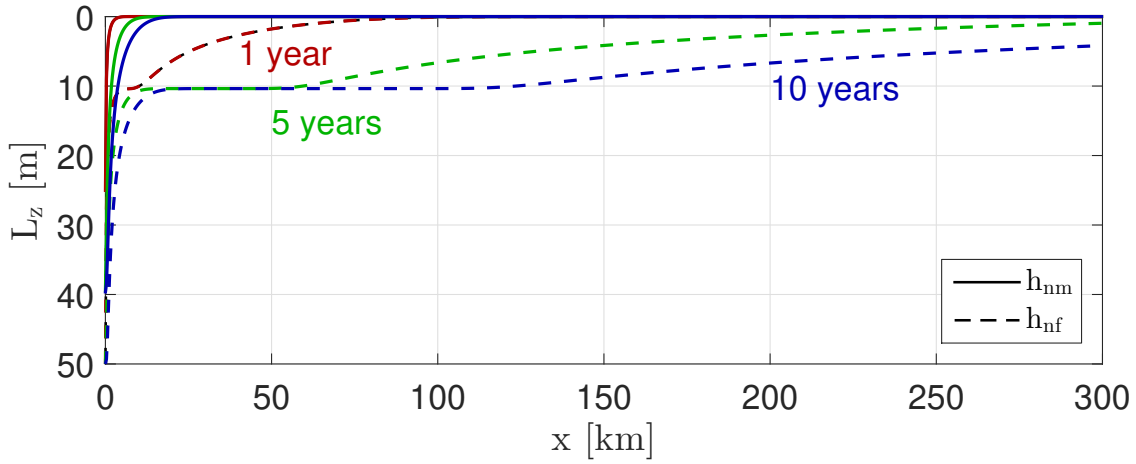

Figure 5.5: CO_2 plumes in the fractures (dashed curves) and matrix (solid curves) for P_e = 75 kPa after 1 year (red curves), 5 years (green curves) and 10 years (blue curves) of injection.

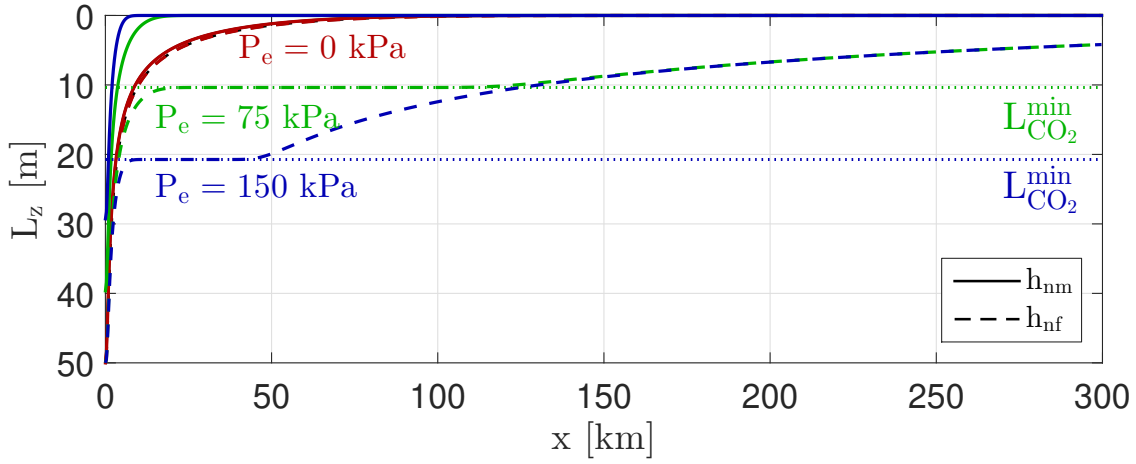

Figure 5.6: CO_2 plumes in the fractures (dashed curves) and matrix (solid curves) for P_e = 0 kPa (red curves), P_e = 75 kPa (green curves) and P_e = 150 kPa (blue curves) after 10 years of injection.

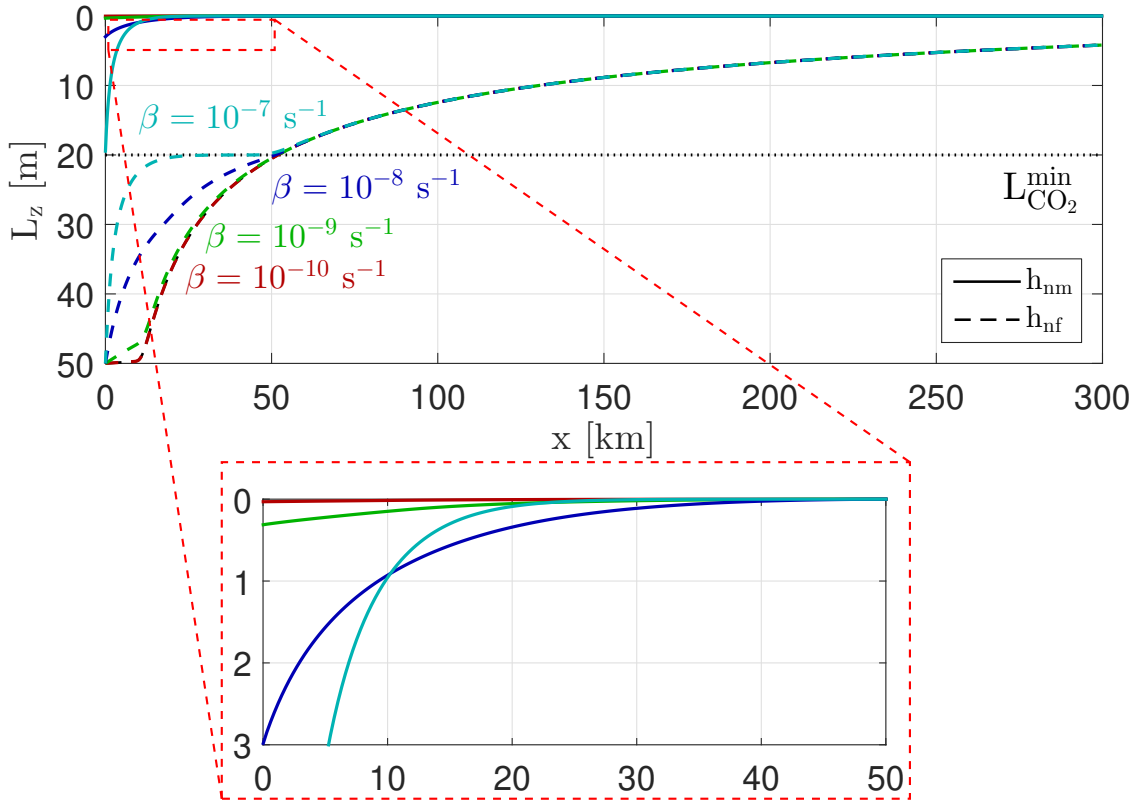

Figure 5.7: CO_2 plumes in the fractures (dashed curves) and matrix (solid curves) for $\beta = 10^{-10}\ s^{-1}$ (red curves), $\beta = 10^{-9}\ s^{-1}$ (green curves), $\beta = 10^{-8}\ s^{-1}$ (blue curves) and $\beta = 10^{-7}\ s^{-1}$ (cyan curves) after 10 years of injection. A zoom-in of the plumes in the matrix is shown.

To assess the suitability of different geological environments, we now analyse the mass of CO_2 that is stored in the matrix after 10 years of injection for twelve simulation cases (Figure 5.8). We consider a fixed transfer-rate coefficient ($\beta = 10^{-6}$ s^{-1}) and consider at M_{CO_2} - the total mass of CO_2 stored in the matrix - for different basin temperatures, P_e's and aquifer depths. M_{CO_2} was obtained

by integrating the CO_2 plume in the matrix. Figure 5.8(a) shows the stored CO_2 in the matrix for the warm basin. Not surprisingly, M_{CO_2} decreases monotonically with an increasing entry pressure. For warm basins, we observe that M_{CO_2} increases with depth, following the increase in ρ_n. For cold basins, however, M_{CO_2} does not increase monotonically with depth (Figure 5.8(b)). We observe that for moderate and high capillary entry pressures, a higher mass of CO_2 is stored in the shallow aquifer in comparison to the deep aquifer. In particular, for P_e = 150 kPa and deep aquifer, a higher M_{CO_2} is observed in the warm basin in contrast with the cold basin. For this particular capillary entry pressure and aquifer depth, a higher basin temperature (and hence lower ρ_n) is more advantageous for storage. This is at first counter-intuitive, as an increase in ρ_n implies an increase in stored mass for unfractured formations.

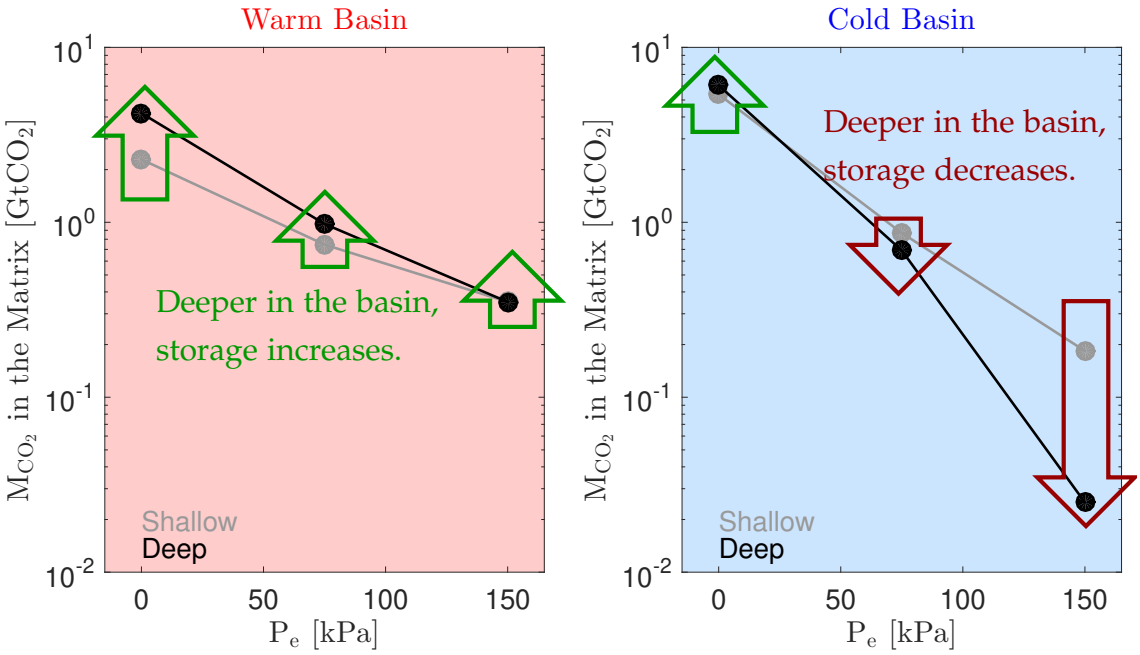

Figure 5.8: Summary of the resulting mass of stored CO_2 obtained from the simulations carried out in this section. M_{CO_2} is plotted for shallow and deep depths in warm (left) and cold basins against capillary entry pressure (P_e).

Figure 5.8 shows that the mass of CO_2 that can be stored in a fractured aquifer depends not only on the temperature and pressure of the formation, but also on the relationship between the density difference and the capillary entry pressure of the matrix, that control the maximum penetration depth of CO_2 in the matrix. Therefore, storage estimates must be redefined for a fractured aquifer. Figure 5.9 shows a depth profile of two important variables that must be taken into account in the estimation of CO_2 storage in a fractured aquifer: the minimum required thickness for CO_2 storage, denoted as $L^{min}_{CO_2}$ (Figure 5.9(a)) and the specific stored mass in the matrix per unit effective area of the aquifer, denoted as $M^a_{CO_2}$ (Figure

5.9(b)). $L^{min}_{CO_2}$ is calculated by setting H - the maximum penetration of CO_2 in the matrix (see Section 3.3) and solving for L:

$$L^{min}_{CO_2} = \frac{P_e}{\Delta\rho g}. \tag{5.10}$$

$L^{min}_{CO_2}$ can be seen as the minimum thickness of the CO_2 plume in the fractures in order to get an infinitesimally small amount of CO_2 in the matrix. $L^{min}_{CO_2}$ has different interpretations depending on the structure of the fracture system and the geological structure of the aquifer. For laterally extensive aquifer with no horizontal fracture sets (or with capillary bridges across vertical stacks of matrix blocks), like the ones considered in this section, $L^{min}_{CO_2}$ is the minimum thickness of the CO_2 plume in the fractures (h_{nf} in Figures 5.5 and 5.6). For structural traps, like the anticlines considered in Chapter 3, $L^{min}_{CO_2}$ is minimum required thickness of the aquifer below the trap, since the filling of the trap will typically happen from top to bottom. If there are horizontal fractures in the aquifer, $L^{min}_{CO_2}$ is the minimum matrix block height. In Figure 5.9(a), we observe $L^{min}_{CO_2}$ for different capillary entry pressures (P_e = 0 kPa, P_e = 75 kPa and P_e = 150 kPa) and for aquifers that lie in warm and cold basins. The solid lines represent P_e = 0, and there is no minimum CO_2 thickness requirement, as there is no capillary force to prevent CO_2 from entering the matrix. Cold basins induce a larger ρ_n and hence smaller $\Delta\rho$ that increases $L^{min}_{CO_2}$. For instance, at a depth of 3000 m and for P_e = 150 kPa, a $L^{min}_{CO_2}$ of 50 m is required to drive CO_2 into the matrix. Therefore, an aquifer of this thickness at this depth and with this capillary pressure should not be considered as a formation with good CO_2 storage potential. The approximate mass of CO_2 that can be stored in an aquifer, $M^a_{CO_2}$ is defined by dividing the mass of CO_2 that is stored in the matrix assuming a CO_2 column of thickness L_{CO_2} by the area of the aquifer and the fraction of the pore volume available for storage

$$\begin{aligned} M_{CO_2} &= \rho_n \overbrace{\left(\phi_m \left(1 - S_{wr}\right)\right)\left(A H_{CO_2}\right)}^{\text{Volume of } CO_2 \text{ in the matrix}} \Rightarrow \\ M^a_{CO_2} &= \frac{M_{CO_2}}{\phi_m \left(1 - S_{wr}\right)\left(A\right)} = \rho_n H_{CO_2}, \end{aligned} \tag{5.11}$$

where $H_{CO_2} = L_{CO_2} - P_e/(\Delta\rho g)$. L_{CO_2} follows the definition of $L^{min}_{CO_2}$, but in the context of the definition of $M^a_{CO_2}$ we see L_{CO_2} as a free parameter. That is, we evaluate the stored mass of CO_2 per area at a given depth for a certain geothermal gradient for a fixed (and arbitrary) value of L_{CO_2}. Figure 5.9(b) shows the depth profile of $M^a_{CO_2}$ for warm and cold basins and L_{CO_2} = 50 m, the aquifer thickness considered for the simulations presented in this section. For P_e = 0 kPa, $M^a_{CO_2}$ behaves as the typical storage estimates for unfractured reservoirs: deeper in the

basin more CO_2 is stored due to higher ρ_n. As the capillary pressure increases, though, we notice that $M^a_{CO_2}$ starts to decrease with depth for cold basins. In particular, for P_e = 150 kPa, after a depth of approximately 1000 m, warm basins are more advantageous for storage than cold basins. After approximately 3000 m, storage in cold basins declines quickly to very low values. This explains the low storage seen for the deep aquifer at a cold basin for P_e = 150 kPa in Figure 5.8.

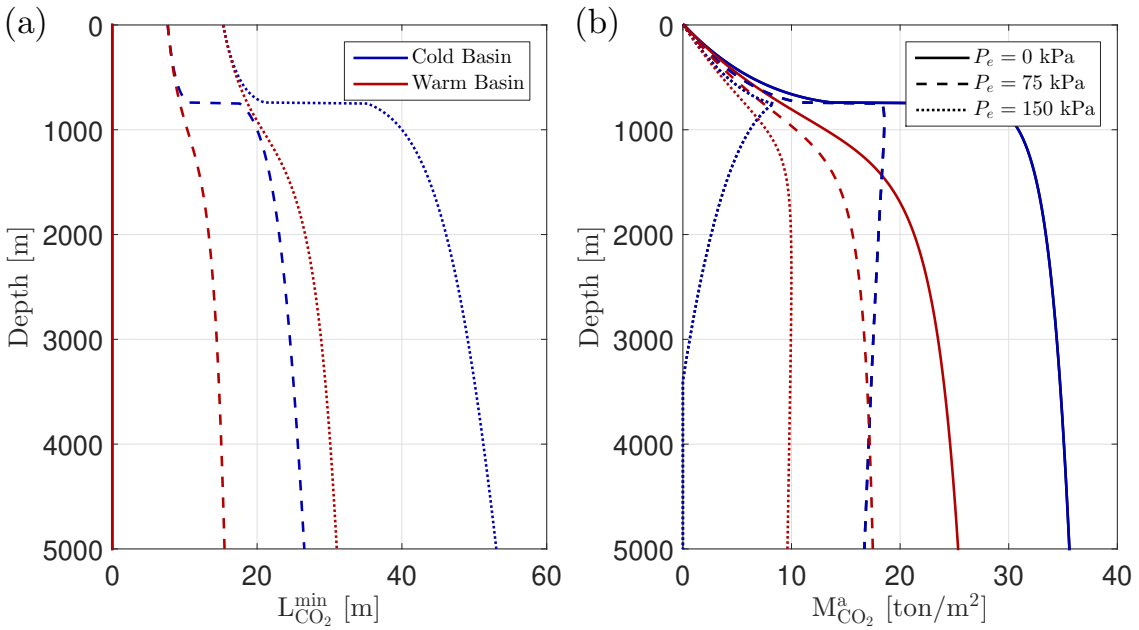

Figure 5.9: Depth profiles of (a) $L^{min}_{CO_2}$ and (b) $M^a_{CO_2}$ for warm basins (red curves) and cold basins (blue curves) for different capillary entry pressures (solid curves correspond to P_e = 0 kPa, dashed curves to P_e = 75 kPa and dotted curves to P_e = 150 kPa).

The findings of this section have implications to storage estimates in NFRs. The two parameters defined above, $L^{min}_{CO_2}$ and $M^a_{CO_2}$, are important to evaluate the CO_2 storage potential of fractured reservoirs. $L^{min}_{CO_2}$ can immediately be used to assess the minimum aquifer thickness or matrix block size in the vertical direction in case there is a set of horizontal fractures. $M^a_{CO_2}$ requires a particular height of the CO_2 plume: L_{CO_2}. For laterally extensive aquifers the VIDP model can be used even when there is no 1D symmetry to estimate the plume thickness (h_{nf}) during the injection period. For fractured anticlines, L_{CO_2} is the thickness of the aquifer below the trap and can be used directly to calculate M_{CO_2}.

5.4 ASSESSMENT OF CO2 STORAGE IN LARGE AQUIFERS

In the previous section, we have evaluated CO_2 storage in NFRs as a function of the aquifer's depth and temperature of the basin, as well as the petrophysical properties. In this section we compare the stored mass of CO_2 in fractured and

unfractured formations using the VIDP model. We consider realistic aquifers whose properties and geological setting were taken from Szulczewski et al. (2012) (aquifer properties shown in Appendix C). The objective of this section is to infer the impact of fractures on geological storage of CO_2.

We base our methodology on the framework proposed by Szulczewski et al. (2012) to estimate CO_2 storage considering limitations imposed by the integrity of the cap rock and by the migration of the CO_2 plume. In this work, a basin-wide simulation of the pressure evolution during CO_2 injection is carried out considering a compressible rock-fluid system and a single phase (brine). This allows the determination of the maximum injection rate that avoid the fracturing of the cap rock. This injection rate is then used in a one-dimensional setting with a vertically-integrated model in order to simulate the plume migration in the aquifer and evaluate storage. Here we apply this framework to a fractured and unfractured version of each aquifer and then compare the storage results (Figure 5.10). This requires a dual-porosity model to evaluate pressure build-up in a fractured formation, which is described in the next subsection.

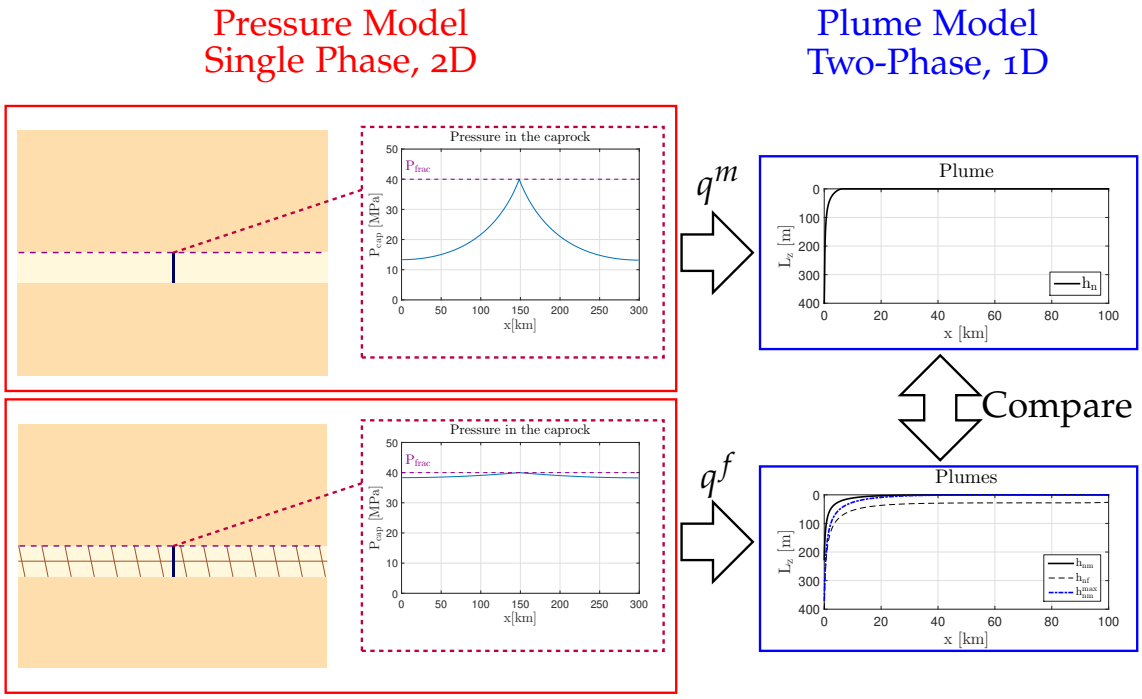

Figure 5.10: Workflow for analysis of CO_2 storage in NFRs. The pressure model (2D simulation with the PressureSimulatorAD code) is used to calculate the maximum allowed injection rate for both fractured and unfractured models. Injection rate then serves as input for the plume migration simulation (1D simulation with the PlumeSimulatorAD). Plumes are integrated at the end of the injection time to provide the final stored mass of CO_2.

5.4.1 *Determination of the Maximum Injection Rate*

The maximum injection rate is estimated by considering a basin-wide simulation of pressure evolution combined with an optimization routine that considers the pressure model as a black box and optimize the injection rate to prevent the pressure at the cap rock to exceed the fracture pressure. The pressure model is based on a mass conservation equation of a single compressible fluid with a source term that represents the well. Boundary conditions and description of the parameters are presented in Figure 5.11. Each aquifer has a thickness L_z, an overburden of thickness D and an underburden of thickness B. Given the uncertainties in the boundaries of the basin, we make the conservative assumption of no-flow boundary conditions for all the boundaries of the pressure model, apart from the top. The top boundary represents the surface and is fixed at atmospheric pressure. The assumption of single-phase flow in the basin simulation means that the compressibility of the CO_2 phase is neglected during the pressure calculation. This is a conservative pressure estimation, as this assumption causes the pressure to be overestimated in the well array. As with the plume model, since injection happens from a line-driven array of wells, some degree of symmetry may be assumed in the model. This time, however, we consider a two-dimensional domain for the pressure computation, since pressure propagates also through the overburden and underburden of the aquifer (vertical direction). Pressure is assumed to be instantaneously equilibrated in the horizontal direction perpendicular to the cross-sectional plane.

In Szulczewski et al. (2012), an equilibration time of 3 years for a well spacing of l = 10 km, a permeability of k = 100 mD, a brine viscosity of μ = 1 cP and an aquifer compressibility of $c = 0.1\ GPa^{-1}$ is estimated. For NFRs the equilibration timescale is expected to be even shorter. For NFRs for which the dual-porosity assumptions are valid, the permeability of the fractures will be typically at least two orders of magnitude higher than the permeability of the matrix. This would lead to an estimated equilibration time of ≈ 0.03 years, which is much smaller than typical injection times.

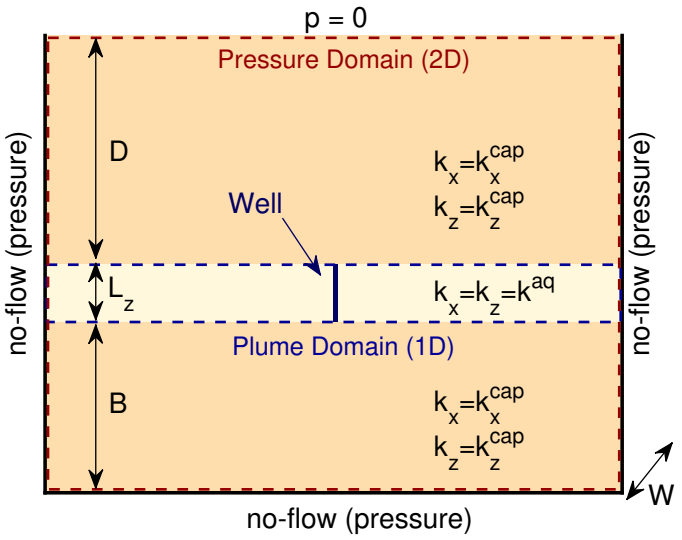

Figure 5.11: Boundary conditions and parameters for the pressure and plume migration models. No-flow boundary conditions are assumed for the lateral and bottom boundaries of the basin model. Zero pressure boundary condition is considered at the top boundary. For the 1D plume migration model, boundaries are considered open for flow.

For an unfractured aquifer, the mass conservation equation in a single continuum with a source term (well) is given by

$$\frac{\partial \phi \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \rho I, \tag{5.12}$$

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu_w} \left(\nabla p - \rho g \nabla z \right), \tag{5.13}$$

$$\rho \left(p \right) = \rho_w e^{cp}, \tag{5.14}$$

where ϕ is the porosity, $\mathbf{k}$ is the permeability, p the brine pressure, I the volumetric injection source and c the total compressibility of the system. Note that $\mathbf{k}$ is anisotropic and heterogeneous, as the aquifer has different permeabilities from the overburden and underburden and the selected aquifers have different permeabilities in horizontal and vertical directions. For fractured aquifers, we solve the dual-porosity dual-permeability equations for single-phase compressible flow, given by

$$\frac{\partial \phi_f \rho}{\partial t} + \nabla \cdot \left(\rho \mathbf{u}_f \right) = \rho I - T_p, \tag{5.15}$$

$$\frac{\partial \phi_m \rho}{\partial t} + \nabla \cdot \left(\rho \mathbf{u}_m \right) = T_p, \tag{5.16}$$

$$\mathbf{u}_f = -\frac{\mathbf{k}_f}{\mu_w} \left(\nabla p_f - \rho g \nabla z \right), \tag{5.17}$$

$$\mathbf{u}_m = -\frac{\mathbf{k}_m}{\mu_w} \left(\nabla p_m - \rho g \nabla z \right), \tag{5.18}$$

where the subscript f denotes properties and fields of the fracture continuum and m denotes properties and fields of matrix continuum. T_p denotes the transfer due to compressibility between both continua. We model the transfer using the first-order model of Kazemi et al. (1976):

$$T_p = \sigma\rho\frac{k_m}{\mu_w}\left(p_f - p_m\right), \tag{5.19}$$

where $\sigma = 4\left(1/a^2 + 1/b^2 + 1/c^2\right)$ is the shape factor as defined by Kazemi et al. (1976). The parameters a, b and c are the fracture spacing (or block size) in the x, y and z directions, respectively. While it is well known that this model will under-predict the transfer during the transient period of pressure propagation in the matrix, we note that the early time will be very short for typical storage sites with high rock permeability. Hence, the error incurred by using a pseudo steady-state model is minimal. Moreover, underestimating the transfer leads to a more conservative estimate in terms of pressure. We consider a fixed fracture spacing of $a = b = 100$ m in the horizontal direction, and $c = L_z$ in the vertical direction (no horizontal fractures). While this choice of a and b is arbitrary, we note that the smaller spacings were reported in the In-Salah CO_2 storage project (Iding and Ringrose, 2010) and seen in fractured outcrops (Egya et al., 2018). A larger fracture spacing is a conservative assumption for pressure build-up estimation. We consider a fixed ratio between fracture and matrix permeabilities, $k_{fi}/k_{mi} = 100$, where subscript i denotes the direction of permeability tensor (x or z). Note that for the fractured pressure model, the dual-porosity dual-permeability equations are solved only in the aquifer. In the overbuden and underburden, we "turn off" the dual porosity behaviour by assigning very low porosities and permeabilities to the matrix (which gives a transfer close to zero) and by assuming that the flowing continuum has the properties of the matrix.

We use the automatic-differentiation framework of MRST to create a class called `PressureSimulatorDP` that runs the pressure model defined by Equations (5.12) - (5.14) or by Equations (5.15) - (5.18), depending on the input provided by the user. The full code is provided in Appendix C. The structure of this class is very similar to the `PlumeSimulatorDP` class described in the Section 5.3. We use the same structures `aquifer` and `dp_info` with fields `continuum1` and `continuum2` to describe the properties of both continua. For the pressure simulator, an additional field of `dp_info` is expected; `coefft`, that is the coefficient that scales the pressure difference in the transfer model T_p in Equation (5.19).

A typical simulation script that runs the 2D pressure model for both unfractured and fractured aquifers is provided in Appendix C. The pressure fields that result from the injection over a period of $T_i = 30$ years in both fractured and unfractured versions of the Mt. Simon (a) aquifer are shown in Figure 5.12. Figure

5.12(a) shows the pressure in the basin at the end of the injection period in the fractured model (top is `continuum1`, which stands for fractures in the aquifer and matrix elsewhere and bottom is `continuum2`, which corresponds to the matrix in the dual-porosity region). When comparing the pressure fields in the fractured model with the pressure in the unfractured model (Figure 5.12(b)), we observe that pressure build-up is much more pronounced in the unfractured model. For this model, in the well region, the cap-rock experiences a pressure as high as approximately 60 MPa, while in the fractured model the pressure in this region is limited to approximately 40 MPa. This is a known and expected result of injection in fractured aquifer: the increased permeability provided by fractures allows for higher injection rates, as pressure diffuses faster in the aquifer instead of building-up quickly in the well region. This increased injectivity can also translate into increased storage with time, compensating the storage loss caused by capillary forces that oppose the flow of CO_2 in the matrix many times.

Notice that the `PressureSimulatorDP` class is a forward simulator of pressure propagation in the basin: given an injection rate, `PressureSimulatorDP` simulates the pressure build-up in the basin. The proposed workflow outlined in Figure 5.10 requires a backward simulation framework, that is, to identify the maximum injection rate that leads to pressures just below the fracture pressure of the cap rock. Therefore, a class named `InjectionRateCalculator` was implemented (see Appendix C), with the goal of encapsulating the `PressureSimulatorDP` class in an optimization procedure. Given an instance of `PressureSimulatorDP`, the class `InjectionRateCalculator` runs pressure simulations as a black box and optimizes $abs\left(max(P_{cap}) - P_{frac}\right)$, where P_{cap} is the pressure at the cap rock, with respect to the injection rate. We use Matlab's `fsolve` function for the optimization. This function uses a subspace trust region method that is based on the interior-reflective Newton method (Mathworks, 2014).

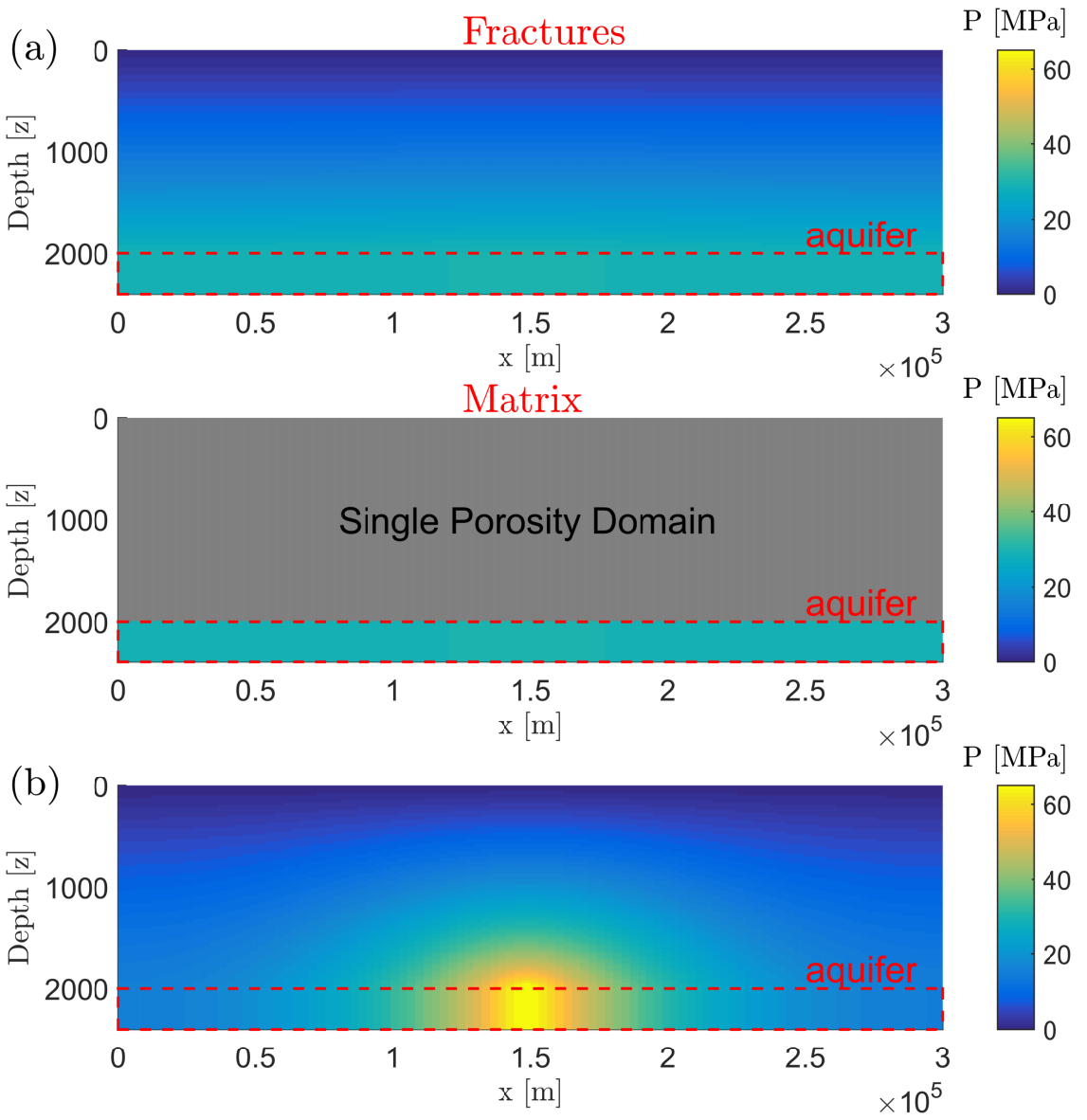

Figure 5.12: Pressure profiles in the Mt. Simon (a) aquifer after 30 years of injection. (a) Fractured model, pressure in the fractures (top) and matrix (bottom). (b) Unfractured model. Fractures accelerate pressure diffusion, which explains higher pressure build-up in the unfractured model.

Figure 5.13 shows the maximum injection rates of the well arrays for the selected aquifers, for both fractured and unfractured models. An injection period of T_i = 30 years is considered here and in the remainder of this chapter. The increase in injection rate (expressed as a percentage) from the unfractured to the fractured model is shown at the top of each bar in the figure. As also observed in Figure 5.12 for the Mt. Simon (a) aquifer, the high permeability of the fracture network allows a higher injectivity in the fractured formations. We observe a wide range of increase in the maximum allowed injection rate (shown above each bar) from approximately 11% to approximately 258%. When we compare the horizontal pressure profiles as seen in Figure 5.10, we observe that fractures "flatten" the pressure curve at the cap rock, preventing the excessive build-up

in the well region. Therefore, fractures will be more helpful for aquifers with a "peaked"(i.e. low injectivity) pressure profile. The shape of the pressure profile is the result of an interplay between the injection rate and the ability of the aquifer, plus underburden and overburden, to diffuse pressure. One of the factors that lead to peaked pressure profiles are low permeabilities of the aquifer and overburden, which explains the low injectivity of Cedar Keys, St. Peter and Mt. Simon (c) aquifers, and also the high increase in injectivity for the fractured models of these aquifers. An exception to this rule is Madison (a). Despite having a very low injectivity for the unfractured model, this aquifer does not show a very high increase in injectivity for the fractured model (only approximately 66%). A possible reason is the fact that the initial pressure of the aquifer, approximately 29 MPa, is very close to the fracture pressure of the cap rock (30 MPa). This leaves little room for improvement in the injectivity. Aquifers with high injectivity, like Potomac (a shallow aquifer with very high permeability) and Navajo-Nugget (a) (an aquifer with very high cap-rock fracture pressure) typically show low values in the injectivity increase. We conclude from Figure 5.13 that in general fractures will lead to reasonable increase of the maximum injection rate. This increase may boost the storage capacity of the fractured formation during the injection period. However, injecting at permissible rate in fractured formations may also lead to fast flow in the fractures and early arrival of the CO_2 plume at the end of the aquifer. Therefore, it may be more favourable to limit the injection rates in order to prevent the early arrival. This concept is explored in the next section where we also consider limited injection rates to compare storage in fractured and unfractured formations.

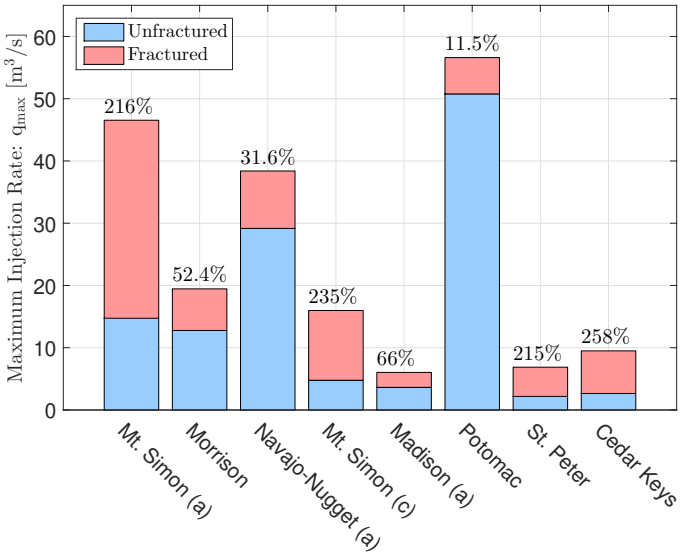

Figure 5.13: Maximum injection rate for the fractured (red) and unfractured (blue) models of each aquifer. Percentage increase caused by fractures in injectivity is seen at the top of each bar.

5.4.2 *Results*

This section presents a comparative study of storage in fractured and unfractured formations, with a focus in assessing the impact of a system of interconnected fractures in CO_2 storage in large aquifers. We use the VIDP model through the code `PlumeSimulatorDP` and determine the maximum injection rate using the code `InjectionRateCalculator`. For each aquifer, we estimate the capillary entry pressure P_e through the relationship between the capillary pressure, interfacial tension and contact angle using the Young-Laplace model:

$$P_e = \frac{2\sigma_{wn} cos\theta_{wn}}{r}, \tag{5.20}$$

where σ_{wn} stands for the interfacial tension between brine and supercritical CO_2, θ_{wn} the contact angle between the phases, and r the characteristic pore size radius, defined as $\sqrt{k_{aq}/\phi_{aq}}$. We interpolate the experimental measurements of interfacial tension at different temperature and pressure conditions provided in Chiquet et al. (2007). Figure 5.14 shows the σ_{wn} points used for the evaluation of P_e. We consider a constant contact angle of $\theta_{wn} = 10\ ^oC$, based on the average value of receding contact angles measured in Saraji et al. (2013). The average aquifer temperature is calculated based on the surface temperature and geothermal gradient of the basin, T_s and G_T. Pressure is calculated assuming a hydrostatic column of saline water. Actual values of P_e, T_s, G_T and ρ_w are provided for each aquifer in Appendix C. The drainage rates β are calculated according to the model defined by Equation (3.9).

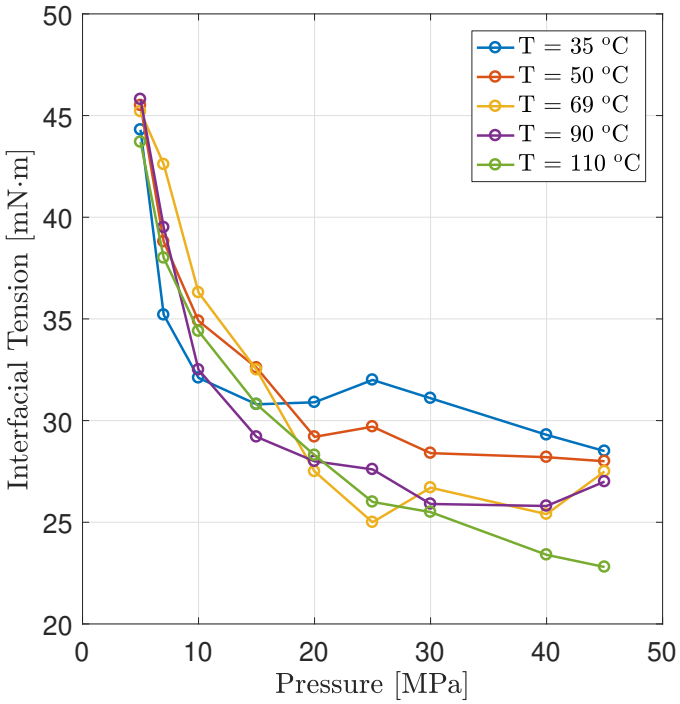

Figure 5.14: CO_2-brine interfacial tension as a function of pressure for different temperatures. Data taken from experiments described in Chiquet et al. (2007)

Three simulations were run for each aquifer:

- Unfractured model: a single-porosity simulation of an unfractured aquifer with pressure-limited injection rates (q^u_{PL}). Injection rate is calculated via the model defined by Equations (5.12) - (5.14) and using the codes `PlumeSimulatorDP` and `InjectionRateCalculator`. Table 5.2 shows the values of q^u_{PL}.

- Fractured model with pressure-limited injection rates: a dual-porosity simulation with pressure-limited injection rates (q^f_{PL}). Injection rate is calculated via the model defined by Equations (5.15) - (5.18) and using the codes `PlumeSimulatorDP` and `InjectionRateCalculator`. Table 5.2 shows the values of q^f_{PL}.

- Fractured model with migration-limited injection rates: a dual-porosity simulation with migration-limited injection rates (q^f_{ML}). Injection rate is reduced from (q^f_{PL}) so that the CO_2 plume tip in the fractures has just net yet reached the boundary of the aquifer at the end of the injection period, that is, there is no CO_2 lost through the boundary. The pressure-limited rates serve as a maximum for calculating the migration-limited rates. Rates are obtained manually by using the code `PlumeSimulatorDP`. Table 5.2 shows the values of q^f_{ML} and ratio between q^f_{ML} and q^f_{PL}.

Aquifer	q^u_{PL} $[m^3/s]$	q^f_{PL} $[m^3/s]$	q^f_{ML} $[m^3/s]$	q^f_{ML}/q^f_{PL}
Mt. Simon (a)	14.73	46.54	23.27	0.5
Morrison	12.76	19.45	9.72	0.5
Navajo-Nugget (a)	29.16	38.38	6.90	0.18
Mt. Simon (c)	4.77	15.98	15.18	0.95
Madison (a)	3.63	6.03	6.03	1.0
Potomac	50.77	56.60	28.30	0.5
St. Peter	2.18	6.87	1.23	0.18
Cedar Keys	2.65	9.49	9.49	1.0

Table 5.2: Injection rates for fractured and unfractured models.

In addition to the CO_2 plumes in fractures and matrix at the end of the injection period, we also analyse two key variables that determine the efficiency of storage in each model, denoted in this subsection as M_{CO_2} and $M^{\%}_{CO_2}$. M_{CO_2} is the mass of CO_2 that is stored in the matrix. $M^{\%}_{CO_2}$ is defined as $M^{\%}_{CO_2} = M^{dom}_{CO_2}/M^{inj}_{CO_2}$, where $M^{dom}_{CO_2}$ is the total mass of CO_2 that is present in the domain (matrix plus fractures for the fractured models) and $M^{inj}_{CO_2}$ is the total mass of CO_2 that is injected by the end of the injection period. $M^{\%}_{CO_2}$ quantifies the amount of CO_2 that saturates the domain; or in other words, the mass that is not lost through the boundaries.

Figure 5.15 shows $M^{\%}_{CO_2}$ plotted against M_{CO_2} for all the models considered in this chapter. We show the final CO_2 plumes and M_{CO_2} with respect to time for four selected cases (St. Peter, Cedar Keys, Madison (a) and Mt. Simon (a)) in Figure 5.16. Many of the aquifers show an expected trend (see, for instance, the Potomac aquifer): the unfractured model has a higher M_{CO_2} with the pressure-limited model coming reasonably close. The high M_{CO_2} for the pressure-limited model is achieved at the cost of spilling some CO_2 through the boundary (note the green square close to the circle, but slightly below the $M^{\%}_{CO_2}$ = 100% line). When injection rates are controlled, buoyancy starts to dominate, which causes the plume to be thinner in the fractures and significantly reduces the buoyant driving force, reducing also M_{CO_2} (note that the green diamond lies below M_{CO_2} = 5 $GtCO_2$). It is interesting to see, however, that most of the pressure-limited rates lead to small spill, and hence, the pressure-limited rates could be explored if there is a reasonable confidence about a sealing aquifer boundary. For Potomac, Morrison, Navajo-Nugget (a) and Mt. Simon (c) aquifers, fractures do not impact significantly the storage (squares are close to the circles for these aquifers and just slightly below the 100% line).

Mt. Simon (a) and Madison (a) are interesting cases that show that fractures can be beneficial for CO_2 storage depending on the geological scenario and rock properties. For the Madison (a) aquifer, simulations show that storage is slightly higher for the fractured models than for the unfractured model. As shown in Figure 5.13, this aquifer has one of the lowest injection rates. Hence, the increase in injectivity caused by the presence of a system of fractures leads to a wider plume in the fracture with an increased storage in the matrix (see Figure 5.16). A deep CO_2 penetration depth (large h_{nm}^{max}, see Equation 5.8), due to large aquifer thickness and high $\Delta\rho$, helps the drainage of the matrix. A similar behaviour is seen for the Mt. Simon (a) aquifer, where the fractured model with pressure-limited rates leads to a storage that is 150% higher than the unfractured model. The migration-limited case shows a stored mass only slightly smaller than the unfractured model, but leads to no spill through the boundary.

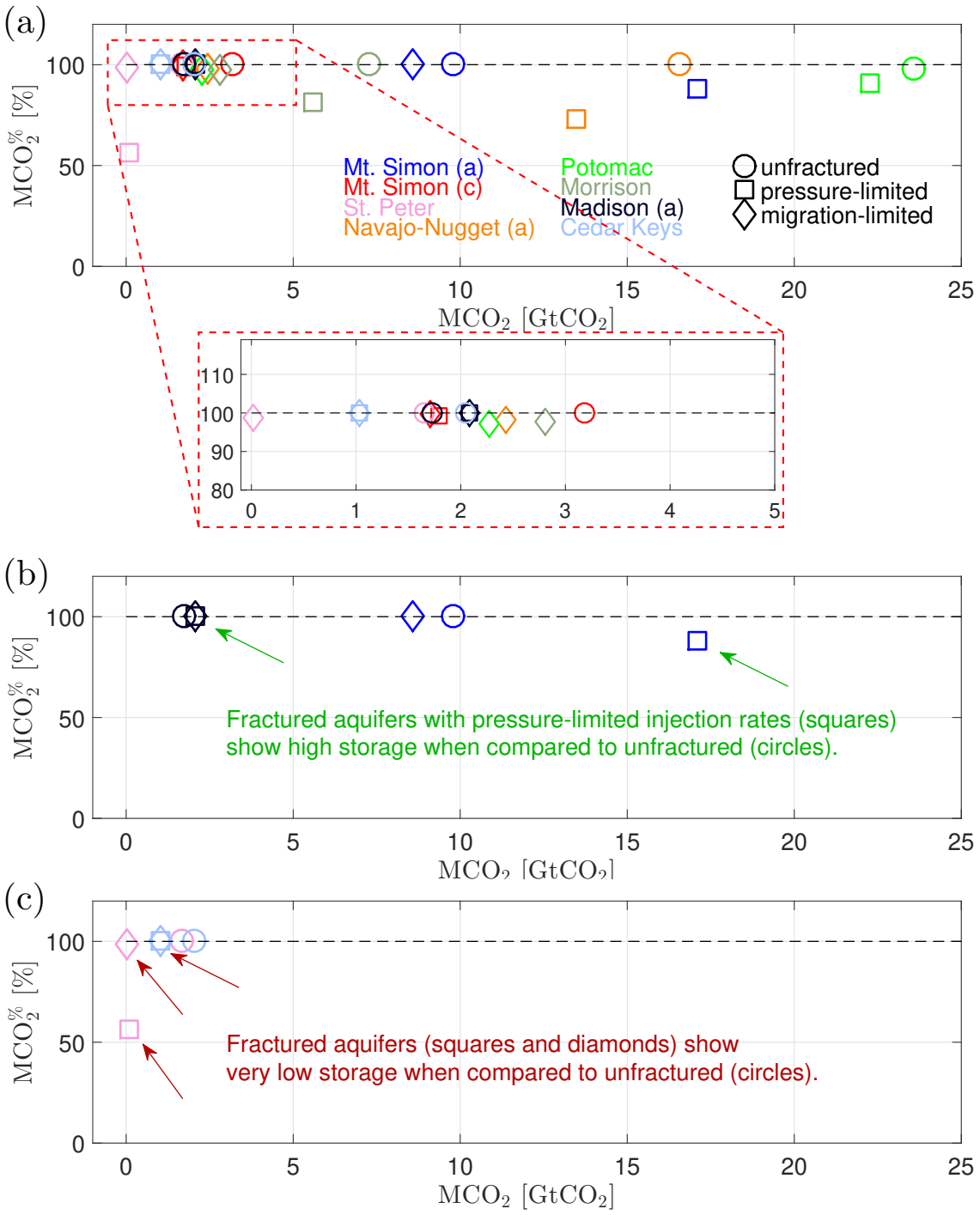

Figure 5.15: (a) Summary of storage results for each case (unfractured, pressure-limited and migration-limited) and each aquifer. $M^{\%}_{CO_2}$ is plotted against M_{CO_2}, the stored mass of CO_2. A zoom-in of the clustered points below M_{CO_2} = 5 $GtCO_2$ is shown. (b) Cases for which fractured aquifers with pressure-limited injection rates show higher storage when compared to unfractured aquifers (Mt. Simon (a) and Madison (a) aquifers). (c) Cases for which fractured aquifers show very low storage when compared to unfractured aquifers (Cedar Keys and St. Peter aquifers).

For two of the eight aquifers considered in this study, fractures affect storage in a very negative way; these are the St. Peter and Cedar Keys aquifers. To understand the poor storage performance of these aquifers, we can calculate the

minimum aquifer thickness necessary to get some CO_2 inside the matrix (see Equation 5.10). We obtain $L^{min}_{CO_2} \approx 129$ m and $L^{min}_{CO_2} \approx 33$ m for Cedar Keys and St. Peter aquifers, respectively. St. Peter aquifer has a thickness of $L_z = 40$ m, which explains why only a negligible amount of CO_2 drains the matrix in these models (see Figure 5.16). Cedar Keys is a thick aquifer ($L_z = 400$ m), but $L^{min}_{CO_2} \approx 129$ m is also a significant thickness that is required for a minimum amount of CO_2 in the matrix. This high value of $L^{min}_{CO_2}$ is caused by a very high capillary entry pressure ($P_e \approx 253$ kPa), due to the low aquifer permeability. Notice that at the end of 30 years of injection, the plume is just below 200 m of depth in the fracture and the CO_2 plume in the matrix is very thin.

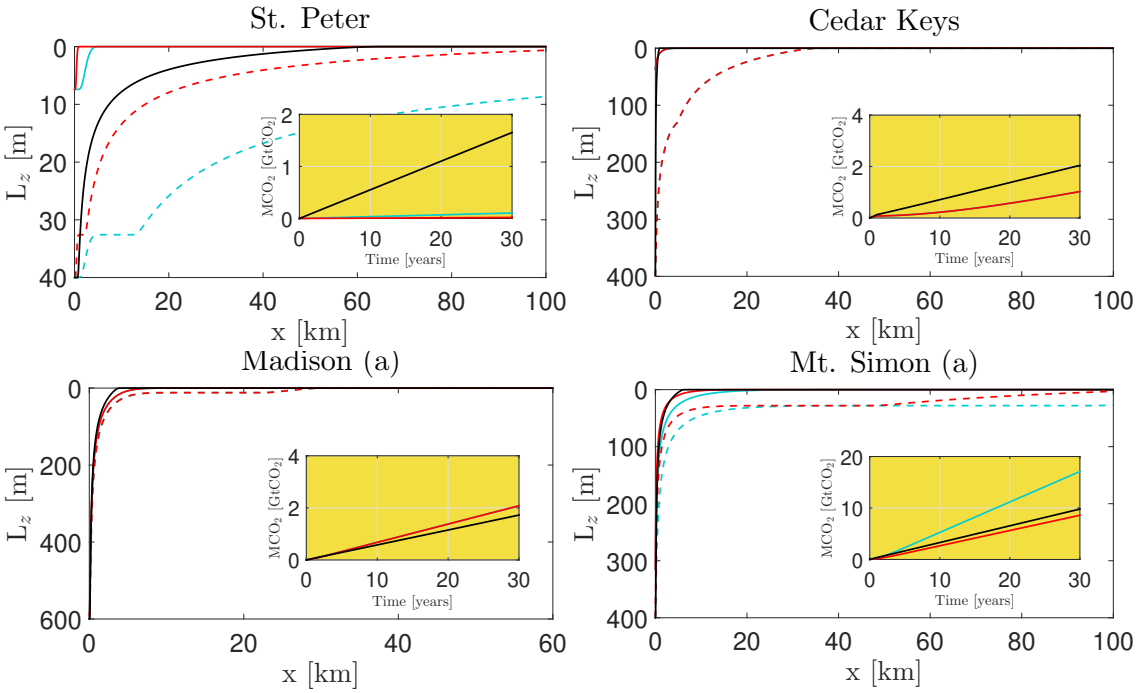

Figure 5.16: CO_2 plumes in the fractures (dashed curves) and in the matrix (solid curves) for the unfractured model (black curves), fractured model with pressure-limited injection rate (turquoise curves) and fractured model with migration-limited injection rate (red curves) for the St. Peter, Cedar Keys, Madison (a) and Mt. Simon (a) aquifers. Time evolution of CO_2 mass in the matrix is shown for each aquifer (yellow inset).

We summarize the impact of fractures in CO_2 storage for the selected aquifers in Table 5.3, where we list the ratio between M_{CO_2} for the unfractured model and M_{CO_2} for each of the fractured models. Factors less than 0.1, representing a storage loss of one order of magnitude, are colored in red and factors higher than 1, representing a positive impact of fractures in storage, are colored in green. We observe that storage in fractured aquifers - contrary to what is generally expected at the moment by the scientific community - will generally be of the same order of magnitude of storage in unfractured aquifers. In some scenarios, the increased injectivity due to the presence of fractures leads to a higher storage in fractured formations. For some of the cases considered here, storage perfor-

mance in fractured aquifers is poor due to the decrease of one or two orders of magnitude in M_{CO_2}. These cases, however were easily identified by exploring the simple concepts defined in the previous section, such as $L^{min}_{CO_2}$.

Aquifer	$M^{pres}_{CO_2}/M^{unf}_{CO_2}$	$M^{migr}_{CO_2}/M^{unf}_{CO_2}$
Mt. Simon (a)	1.74	0.87
Morrison	0.77	0.38
Navajo-Nugget (a)	0.81	0.14
Mt. Simon (c)	0.56	0.53
Madison (a)	1.20	1.20
Potomac	0.94	0.09
St. Peter	0.06	0.01
Cedar Keys	0.50	0.50

Table 5.3: Ratio between stored mass of CO_2 in the fractured and unfractured models for the pressure-limited injection rate (second column) and migration-limited injection rate (third column). For most of the cases considered here, order of magnitude of stored mass is not changed if a system of fractures exists in the formation.

5.5 SUMMARY AND CONCLUSIONS

This chapter aimed at answering a specific yet very relevant question: how detrimental are fractures to CO_2 storage ? The general prevailing understanding at the moment is that a system of interconnected fractures would make CO_2 storage unfeasible due to very early breakthrough of the CO_2 plume and low drainage rates and maximum CO_2 in the matrix due to capillary forces. In this chapter, we have shown that this is not necessarily the case. In fact, fractures can even boost storage - if we consider the plume migration - due to increased injectivity.

In order to achieve the results, a novel vertically-integrated model with dual-porosity behaviour was proposed and implemented in the MRST-AD framework. We have used the drainage transfer function that was developed in Chapter 3 and improves significantly the representation of the physics in comparison to other models available in the literature. We have validated the model against single-porosity vertically-integrated results available in the literature with very good agreement.

This model was used first to analyse the sensitivity of storage to several geologic settings; in particular, temperature and pressure of the formation and its capillary pressure. It was shown that temperature and pressure affect storage in a more intricate way than it does for unfractured formations. An increase in CO_2 density causes the decrease of $\Delta\rho$, which in turn may cause a decrease in storage

due to the decreased buoyancy driving force. Two important parameters to measure storage potential in NFRs were defined: $L^{min}_{CO_2}$ and $M^{a}_{CO_2}$. These parameters help ranking fractured aquifers according to the storage potential.

Finally, we investigated CO_2 storage in realistic aquifers considering the (hypothetical) existence of an interconnected system of fractures. We have developed another AD-based simulator to obtain the maximum injection rates in fractured and unfractured formations. Fracture increase significantly the injectivity of CO_2, what boosts storages for some aquifers. When comparing storage in fractured and unfractured formations, we have seen that storage with and without fractures is almost always comparable. For some aquifers, fractured models show better storage performance when compared to the unfractured models.

This chapter provided tools for quick assessment of storage potential in large fractured aquifers. These tools were used to investigate the impact of fractures in real aquifers taken from the literature, with interesting findings. Although fractures will typically lead to a decrease in storage, this decrease will not generally exceed one order of magnitude. Therefore, large fractured aquifers seem to be suitable for storage and should be evaluated on a case-by-case basis.

6

SUMMARY, CONCLUSION AND GUIDELINES FOR CO_2 STORAGE IN NFR'S

Carbon capture and storage (CCS) is a promising technology to reduce the concentration of greenhouse gases in the atmosphere. Fractured reservoirs are ubiquitous across the world. A large percentage of the world's hydrocarbon resources come from naturally fractured reservoirs that could be later used for CO_2 storage. However, fractured reservoirs are not typically considered as formations that have potential to safely store CO_2. Although considerable efforts are under way to assess CO_2 storage potential in carbonate formations which are typically fractured (Alhammadi et al., 2017; Menke et al., 2018; Al-Khulaifi et al., 2018), to date, the impact of fractures is largely ignored. The main concerns regarding storage in fractured formations are the high conductivity of the fracture system, which may lead to loss of control on the CO_2 plume, and capillary forces that will tend to keep CO_2 in the fractures, that typically have very low pore volume and do not provide good storage potential. Simulating the dynamics in naturally fractured reservoirs is a complex and computationally-intensive task, as there are several mechanisms by which fractures and matrix exchange fluids. Moreover, fractures introduce a new timescale to fluid flow, as the transfer to the matrix will typically be much slower than the flow in the fractures. Yet, numerical simulations are extremely useful to obtain insights about the reservoir dynamics during CO_2 storage in fractured formations.

Dual-porosity models have been used for many years due to their simplicity and computational efficiency. They use an upscaled representation of the transfer behaviour and averaged properties of the fracture system. The transfer behaviour is represented by transfer functions that describe the fluid exchange between fractures and matrix. In CO_2 storage operations, the main physical mechanisms responsible for fracture-matrix fluid exchange are gravity drainage, during CO_2 injection, and spontaneous imbibition during CO_2 plume migration after injection (Chapter 1). During the course of this thesis, we have identified that existing formulations that model these transfer mechanisms fail to capture the physics of the transfer accurately. Therefore, we have developed new physical models that are able to model gravity drainage (Chapter 3) and spontaneous imbibition

(Chapter 4) in matrix blocks and are particularly fit for CO_2 storage applications. We have also implemented a new dual-porosity module for MRST (Chapter 2) that allows us to test the new transfer functions in dual-porosity models of NFRs.

The gravity drainage transfer function was then applied to a field scale to understand storage and plume migration in fractured aquifers. First an anticline model (Chapter 3) and then large horizontal aquifers (Chapter 5) were investigated. In the latter, we have also developed important tools for the assessment of CO_2 storage potential, namely a vertically-integrated model and simple metrics that allow us to rank fractured formations and to redefine storage potential. By applying these tools to representations of real aquifers from the literature, we found that the impact of fractures on the fast flow of the CO_2 plume is "manageable" by controlling injection rates. Moreover, we found that the increased injectivity due to the presence of fractures boosts storage and minimizes the storage loss caused by fractures.

The findings of this thesis allow us to to provide general guidelines for CO_2 storage in NFRs:

- **To correct storage estimates to account for a loss in storativity caused by the presence of a fracture system.** It is common to estimate the storage potential for unfractured formations using simple volumetric estimates that calculate the pore volume of the formation and multiply by an average density of CO_2 at depth. In Szulczewski et al. (2012), the authors have challenged this approach and have shown that it is important to consider plume migration and pressure limitations in order to estimate storage in saline aquifers correctly. They suggest the the migration-limited storage capacity, C_t, is given by:

$$C_t = \rho_n L_t W L_z \phi \left(1 - S_{wr}\right) \frac{2}{\xi_T}, \tag{6.1}$$

  where ξ_T is the dimensionless position of the fully-trapped plume tip. The dimensionless position ξ_T is taken from 1D simulation of vertically-integrated plume migration model. While we did not consider capillary trapping here, we have taken inspiration from this model in order to define an estimation of storage capacity for NFRs. The estimated is based on the amount of CO_2 that drains the matrix at the end of the injection period T_i and is given by

$$M_{CO_2} = \rho_n W \phi \left(1 - S_{wr}\right) 2 \int_0^{L_t} \left(L_z - h_{nm}\left(x, T_i\right)\right) dx, \tag{6.2}$$

  where the integral is evaluated numerically. While this metric requires the utilization of the vertically-integrated dual-porosity code, it is essential to

capture the order of magnitude of storage in NFRs. The utilization of a volumetric estimate without considering plume thickness and migration may lead to severe overestimation of the storage, since the plume in the fracture may not be sufficient to allow the drainage of the matrix.

- **To use of the minimum plume height ($L^{min}_{CO_2}$) and the stored mass ($M^{a}_{CO_2}$) for quick evaluation of storage potential.** These two simple variables allow us to rank aquifers with respect to CO_2 storage potential. $L^{min}_{CO_2}$ can be compared to the aquifer thickness. $M^{a}_{CO_2}$ denotes the mass per unit area and requires a plume thickness in the fractures. For fractured anticlines, like the ones evaluated in Chapter 3, $M^{a}_{CO_2}$ can be immediately used to estimate storage, as the final thickness of the CO_2 plume in the fractures will be necessarily the thickness below the structural trap. For large horizontal or sloping aquifers, the vertically-integrated dual-porosity model must be run in order to estimate storage.

- **To apply the gravity-drainage transfer function to evaluate the spill time and to design an optimal injection plan for the fractured aquifer.** The model given by Equation (3.14) gives an order of magnitude of the maximum injection rate that avoids early spill of CO_2. However, this model must be used with caution as it does not consider geometrical aspects of the formation. Numerical simulations with the gravity drainage transfer function, either with a fully 3D numerical simulator or with the VIDP model, could be carried out to have more precise estimates of the maximum allowed injection rate.

The developments in this thesis provide a simplified first-order physics analysis of storage in fractured formations and plume migration in the fractures. We believe they are an important initial step in the analysis of CO_2 storage in NFRs. The results presented in this thesis show that in principle fractured formations have storage potential but require the injection rates to be managed to avoid the early-spill, and pose additional geological constraints. Thus, NFRs should be evaluated on a case-by-case basis instead of simpy being disregarded as candidates for geological storage of CO_2.

6.1 SUGGESTION OF FUTURE DEVELOPMENTS IN THE FIELD

The following points are natural future developments of the work presented in this thesis.

- **Investigation of the effect of geomechanics in storage and matrix-fracture transfer.** Coupling of fluid flow with geomechanics for fractured reservoirs

is numerically very challenging and has been object of extensive research in the past few years (Rutqvist et al., 2010; Rutqvist, 2012; Verdon et al., 2013). This thesis has neglected effects of geomechanics in fracture opening/closing or in the transfer. However, it is well known that geomechanics may play a very important role during carbon storage operations. In fractured reservoirs, stress changes during injection can cause fractures to open or close and matrix blocks to shrink or expand. On the one hand, during injection of CO_2, matrix block shrinkage may lead to transfer of brine to the fractures, increasing the extent of the two-phase region and retaining the advancement of the CO_2 plume. On the other hand, fracture opening during injection of CO_2 will increase the fracture transmissibility and hence make the CO_2 plume travel even faster. This could be an interesting "self-cancelling" effect of geomechanics during CO_2 injection that should be investigated more carefully. It opens, though, a whole new spectrum of new problems that need to be solved, such as the development of reliable models for stress-based effective permeabilities of fractures and the development of a transfer function that couples drainage with geomechanics.

- **Extension of the spontaneous imbibition transfer function for 2D and 3D settings.** The transfer function developed in Chapter 4 is suitable for a one-dimensional setting; that is, when there is only one set of nearly-parallel fractures. This situation occurs, for instance, in the the In-Salah field. If the aquifer has two or three sets of fractures, matrix block shapes are 2D or 3D and are not automatically handled by the transfer function developed in this thesis. The idea presented here, however, could be extrapolated to higher dimensions where approximate analytical solutions exist for the early-time behaviour and the exponential model can still be used for the late time. The transition timescale - denoted in this thesis as $\hat{t}$ - will change due to interference between fronts. Therefore, a consistent analysis must be carried out to understand what are the key properties/groups affecting this timescale and how to predict it based solely on rock and fluid properties like it was done for the 1D case.

- **Investigation of capillary trapping.** One of the main mechanisms for safe trapping of CO_2 is capillary trapping (Juanes et al., 2006). In this trapping mechanism, imbibition of brine into a CO_2 saturated zone generates the snap-off of the CO_2 phase, trapping it as immobile phase in the pores of the rock. Most of this thesis was devoted to understanding under what conditions CO_2 is driven into the rock matrix. A natural next step - not investigated here - is to understand the trapping of CO_2 in the matrix when the water level rises after plume migration to the cap-rock. The CO_2 saturated blocks, now surrounded by brine, will be imbibed and this will

lead to trapping, although it will also lead to the flow of CO_2 back to the fractures. The imbibition transfer function developed here is interesting because it provides a procedure for reconstructing the saturation profile inside the matrix, an information that is key to determine trapping with precision.

- **Formulation of the VIDP model in radial coordinates.** We have developed a one-dimensional vertically-integrated model for simulating plume migration in linear coordinates. This model is useful when there is a linear 1D symmetry, that is, when injection happens in large aquifer with a line of wells with small spacing between them. When there are few wells in the aquifer, it may be more convenient to have a model that relies on radial symmetry, which may be achieved by formulating the VIDP in radial coordinates.

- **Improvement of the model for estimation of the maximum injection rate.** The simple, albeit effective model provided in Chapter 3 for estimating the maximum injection rate that avoids early spill should be improved to include information about the geometry of the aquifer and of the CO_2 plume. Currently, the model assumes that a plume with a volume matching the volume below the anticline is "instantaneously" placed in the aquifer and consider the time it takes for this amount of CO_2 to be transferred to the matrix. Despite being simplistic, it seems to approximate the order of magnitude of the optimal injection rates correctly, as shown in Chapter 3. More research is required to develop a model that is more precise but still does not require numerical simulations for the determination of the maximum injection rate.

A

SUPPORTING INFORMATION OF CHAPTER 1

A.1 DATA FOR SIMULATION OF JOHANSEN FIELD

Property	Value	Unit	**Property**	Value	Unit
k_x^m	405	mD	$k_{rn}^{max,f}$	1	-
k_z^m	405	mD	$k_{rw}^{max,f}$	1	-
k_x^f	4053	mD	S_{wr}^m	0.42	-
k_z^f	4053	mD	S_{wr}^f	0	-
ϕ^m	0.19	-	n_w^m	1.7	-
ϕ^f	0.005	-	n_n^m	2.8	-
μ_w	0.8	cP	n_w^f	1	-
μ_n	0.05	cP	n_n^f	1	-
$k_{rn}^{max,m}$	0.26	-	ρ_w	1012	kg/m^3
$k_{rw}^{max,m}$	1	-	ρ_n	714	kg/m^3
P_e	5	kPa	n_p	0.5	-

Table A.1: Data used in the Johansen field simulation in Chapter 1. The geological model is distributed with MRST in the co2lab module. A matrix block of size $a = b = c = 10$ m was considered. "Physically-based transfer function" corresponds to the one developed in Chapter 3 (Equation (3.11)) with $\beta = 3.48 \times 10^{-8}$ and $S_{nm}^{\max} = 0.395774$. The parameters β and $S_{nm}^{\max}$ were calculated based on the parameters presented in this table and according to the theory developed in Chapter 3. "Conventional transfer function" corresponds to the Gilman-Kazemi transfer function (Chapter 1, Equation (1.10)) with the Kazemi et al. (1976) shape factor (see Table 1.1).

B

SUPPORTING INFORMATION OF CHAPTER 2

B.1 ECLIPSE FILE FOR GRAVITY-DRAINAGE BENCHMARK

```
RUNSPEC
TITLE
CO2 Block Simulation V2

DIMENS
   1   1   2 / -- Two cells on x direction and 1 on z direction. Second
       layer is fracture

DUALPORO

OIL

GAS

METRIC

TABDIMS
    2    1   20   20    2   20 /

GRAVDR

REGDIMS
    2    1    0    0  /

START
   16 'JUN' 2017  /

UNIFOUT

UNIFIN
```

```
GRID
INIT

NODPPM

-- USED FOR GRAVITY DRAINAGE CALCULATIONS

DZMTRXV
 1*10.0/


DPGRID

EQUALS                                          FIELD   09:47  1 AUG 88
'TOPS    '  1280.  ,  1, 1  ,  1,  1  ,  1,  1  /  MATRIX
'DX      ' 5.000000  ,  1, 1  ,  1,  1  ,  1,  2  /
'DY      ' 5.00000  ,   1, 1  ,  1,  1  ,  1,  2  /
'DZ      ' 10.0000000  ,  1, 1  ,  1,  1  ,  1,  2  /
'PERMX   ' 15.00000000  ,  1, 1  ,  1,  1  ,  1,  1  /
'PERMY   ' 15.00000000  ,  1, 1  ,  1,  1  ,  1,  1  /
'PERMZ   ' 15.00000000  ,  1, 1  ,  1,  1  ,  1,  1  /
'PORO    '  .15000000  ,  1, 1  ,  1,  1  ,  1,  1  /
'PORO    '  .015000000  ,  1, 1  ,  1,  1  ,  2,  2   /  FRACTURE
'PERMX   ' 100000.0000000  ,  1, 1  ,  1,  1  ,  2,  2  /
'PERMY   ' 100000.0000000  ,  1, 1  ,  1,  1  ,  2,  2  /
'PERMZ   ' 100000.0000000  ,  1, 1  ,  1,  1  ,  2,  2  /
'MULTPV  ' 500000.0000000  ,  1, 1  ,  1,  1  ,  2,  2 /
/

RPTGRID                                         FIELD   14:29  5 AUG 88
      /

SIGMAV                                          FIELD   09:59  1 AUG 88
  1*0.3600/


PROPS

INTPC

'BOTH'
/
```

```
--SWOF       2 TABLES   20 NODES IN EACH          FIELD   11:55  1 AUG 88
--         0         0    1.0000    0.7906
--    0.1000    0.1000    0.9000    0.7906
--    0.2000    0.2000    0.8000    0.5590
--    0.3000    0.3000    0.7000    0.4564
--    0.4000    0.4000    0.6000    0.3953
--    0.5000    0.5000    0.5000    0.3536
--    0.6000    0.6000    0.4000    0.3227
--    0.7000    0.7000    0.3000    0.2988
--    0.8000    0.8000    0.2000    0.2795
--    0.9000    0.9000    0.1000    0.2635
--    1.0000    1.0000         0    0.2500
--/
--0.  0.  1.  0.
--1.  1.  0.  0.
--/

SGOF      2 TABLES   20 NODES IN EACH          FIELD   11:55  1 AUG 88
         0         0    1.0000    0.0500
    0.1000    0.1000    0.9000    0.0527
    0.2000    0.2000    0.8000    0.0559
    0.3000    0.3000    0.7000    0.0598
    0.4000    0.4000    0.6000    0.0645
    0.5000    0.5000    0.5000    0.0707
    0.6000    0.6000    0.4000    0.0791
    0.7000    0.7000    0.3000    0.0913
    0.8000    0.8000    0.2000    0.1118
    0.9000    0.9000    0.1000    0.1581
    1.0000    1.0000         0    0.1581
/
0.  0.  1.  0.
1.  1.  0.  0.
/

PVDG
60.000 1.0001 .04001
70.000 1.0000 .04000
/

ROCK       1 TABLES   20 P NODES   20 R NODES    FIELD   10:41  1 AUG 88
 60.00      1E-12 /
```

```
PVDO       1 TABLES   20 P NODES   20 R NODES    FIELD   12:00  1 AUG 88
 60.0 1.000001  1.0
 70.0 1.000000  1.00001
/

DENSITY    1 TABLES   20 P NODES   20 R NODES    FIELD   10:43  1 AUG 88
 1000  1200  600 /

/

RPTPROPS                                        FIELD   14:29  5 AUG 88
      /


REGIONS

EQUALS                                          FIELD   17:22  4 AUG 88
'SATNUM  ' 1           ,  1, 1  ,  1,  1  ,  1,  1  /
'SATNUM  ' 2           ,  1, 1  ,  1,  1  ,  2,  2  /
/

FIPNUM                                          FIELD   17:23  4 AUG 88
 1*1 1*2 /


SOLUTION

SGAS
1*0
1*1 /

PRESSURE
1*60
1*60 /

RPTSOL
-- Initialisation Print Output
--
'PRES' 'SOIL' 'SWAT' 'SGAS' 'RS' 'PWAT' 'PGAS'   /


SUMMARY
```

```
BGSAT
1 1 1 /
1 1 2 /
/

EXCEL

SCHEDULE

RPTSCHED
'PRES' 'SOIL' 'SWAT' 'SGAS' 'RS' 'RESTART=2' 'SUMMARY=1' 'CPU=1' 'PWAT'
'PGAS' /

TSTEP                                          FIELD   11:05  1 AUG 88
 90*1
/
END
```

C

SUPPORTING INFORMATION OF CHAPTER 5

C.1 AUTOMATIC-DIFFERENTIATION CODES

Example script for running simulation with `PlumeSimulatorDP.m`

```
%% Aquifer Parameters
aquifer = mtsimonaAquifer();

%% Injection time
Ti = 10*year;

%% dp info
shape_factor = 0.01;
kt = aquifer.kaqx;
mut = aquifer.muw;

dp_info_dp.continuum1.kx = 100*aquifer.kaqx;
dp_info_dp.continuum1.kz = 100*aquifer.kaqz;
dp_info_dp.continuum1.phi = 0.01;

dp_info_dp.continuum2.kx = aquifer.kaqx;
dp_info_dp.continuum2.kz = aquifer.kaqz;
dp_info_dp.continuum2.phi = aquifer.phi;

% Setting beta and capillary entry pressure
dp_info_dp.Pe = 100*kilo*Pascal;
dp_info_dp.beta = 1e-08;

%% Mesh
Nx = 10000;

%% Maximum Injection rate
qinj = 1.7064e-04;
```

```
%% Setting up pressure simulation
ps_dp = PlumeSimulatorDP(aquifer,dp_info_dp);
ps_dp = ps_dp.setupSimulation(Nx);

%% Initialize AD variables
h1_init_dp = zeros(ps_dp.G.cells.num,1);
h2_init_dp = zeros(ps_dp.G.cells.num,1);
hcell_dp = {h1_init_dp,h2_init_dp};

%% Simulation parameters
totTime  = Ti;             % total simulation time
dt       = 1*year;  % constant time step
tol      = 1e-5;                % Newton tolerance
maxits   = 10;                  % max number of Newton its

%% Time integration
[hcell_dp] = ps_dp.integrateTime(hcell_dp,dt,totTime,qinj);
```

Example script for running simulation with `PressureSimulatorDP.m`

```
%% Aquifer Parameters
aquifer = mtsimonaAquifer();

%% Injection time
Ti = 30*year;

%% Defining single porosity model
clear dp_info_sp;
dp_info_sp.continuum1.kx = aquifer.kaqx;
dp_info_sp.continuum1.kz = aquifer.kaqz;
dp_info_sp.continuum1.phi = aquifer.phi;

%% Defining dual porosity model
clear dp_info_dp;
Lx = 100;
Ly = 100;
Lz = aquifer.H;
shape_factor = 4*(1/Lx^2+1/Ly^2+1/Lz^2);
kt = aquifer.kaqx;
mut = aquifer.muw;

dp_info_dp.continuum1.kx = 100*aquifer.kaqx;
dp_info_dp.continuum1.kz = 100*aquifer.kaqz;
```

```
dp_info_dp.continuum1.phi = 0.01;

dp_info_dp.continuum2.kx = aquifer.kaqx;
dp_info_dp.continuum2.kz = aquifer.kaqz;
dp_info_dp.continuum2.phi = aquifer.phi;
dp_info_dp.coefft = shape_factor*(kt/mut);

%% Initializing and preparing the single porosity model
pres_sp = PressureSimulatorDP(aquifer,dp_info_sp);
pres_sp = pres_sp.setupSimulation(100,100);

%% Initializing and preparing the dual porosity model
model_type = 'dpdk';
pres_dp = PressureSimulatorDP(aquifer,dp_info_dp,model_type);
pres_dp = pres_dp.setupSimulation(100,100);

%% Initializing pressure field
g = 9.81;
p_init = aquifer.rhow * g * pres_dp.G.cells.centroids(:,3);

pf = p_init;
pm = p_init;
pcell_sp = {pf,pm};
pcell_dp = {pf,pm};

%% Source term
Q = 10; % injection rate of the well array
Is = zeros(pres_dp.G.cells.num,1);
Is(pres_dp.cell_groups.well_cells) = Q/aquifer.W/...
                        length(pres_dp.cell_groups.well_cells);

dt = 1*year;
[pcell_sp] = pres_sp.integrateTime(pcell_sp,dt,Is,Ti,1);
[pcell_dp] = pres_dp.integrateTime(pcell_dp,dt,Is,Ti,1);
```

Class PressureSimulatorDP.m

```
classdef PressureSimulatorDP
% Class PressureSimulatorDP

properties
        aquifer
        dp_info
        model_type
        operators
        equations
        cell_groups
        G
        rock
        rock_matrix
end

methods

function [ps] = PressureSimulatorDP(aquifer,dp_info,model_type)
        ps.aquifer = aquifer;
        ps.dp_info = dp_info;
        if(nargin<3)
                ps.model_type = 'dp';
        else
                ps.model_type = model_type;
        end
end

function [ps] = setupSimulation(ps,Nx,Nz)

        %% Mesh
        dz = (ps.aquifer.D+ps.aquifer.H+ps.aquifer.B)/Nz;

        %% Set up model geometry
        [nx,ny,nz] = deal( Nx,  1, Nz);
        [Dx,Dy,Dz] = deal(ps.aquifer.L,1,ps.aquifer.D + ps.aquifer.H + ps.
            aquifer.B);
        ps.G = cartGrid([nx, ny, nz], [Dx, Dy, Dz]);
        ps.G = computeGeometry(ps.G);

        %% Marking aquifer and well cells
        closest_dx = min(abs(ps.G.cells.centroids(:,1)-ps.aquifer.xmin));
```

```
ps.cell_groups.well_cells = find(abs(ps.G.cells.centroids(:,1)-(ps.
    aquifer.xmin - closest_dx))<=0.00001 &  ps.G.cells.centroids
    (:,3)>=ps.aquifer.D & ps.G.cells.centroids(:,3)<=ps.aquifer.D +
    ps.aquifer.H);

ps.cell_groups.aquifer_cells = find(ps.G.cells.centroids(:,3)>=ps.
    aquifer.D & ps.G.cells.centroids(:,3)<=ps.aquifer.D + ps.aquifer
    .H);

closest_dz = min(abs(ps.G.cells.centroids(:,3)-ps.aquifer.D));

ps.cell_groups.cap_rock_cells = find(abs(ps.G.cells.centroids(:,3)
    ...
                                                    -(ps.aquifer.D -
                                                        closest_dz))
                                                        <=0.00001);

if(isempty(ps.cell_groups.cap_rock_cells))
        ps.cell_groups.cap_rock_cells = find(abs(ps.G.cells.
            centroids(:,3)-(ps.aquifer.D + closest_dz) <=0.00001);
end

%% Define rock model
ps.rock.poro = ps.aquifer.phicap*ones(ps.G.cells.num,1);
ps.rock.perm =[ps.aquifer.kcapx*ones(ps.G.cells.num,1),ps.aquifer.
    kcapx*ones(ps.G.cells.num,1),ps.aquifer.kcapz*ones(ps.G.cells.
    num,1)];

ps.rock.perm(ps.cell_groups.aquifer_cells,1:2) = ps.dp_info.
    continuum1.kx;
ps.rock.perm(ps.cell_groups.aquifer_cells,3) = ps.dp_info.continuum
    1.kz;
ps.rock.poro(ps.cell_groups.aquifer_cells) = ps.dp_info.continuum1.
    phi;

pv = poreVolume(ps.G, ps.rock);

if(isfield(ps.dp_info,'continuum2'))
        %% If there is a second continuum:
        %% -- in the aquifer: continuum1 = fraccture, continuum2 =
            matrix
        %% -- in the over/under burden: continuum1 = matrix,
            transfer = 0
```

```
        ps.rock_matrix.poro = 0.000001*ones(ps.G.cells.num,1);
        ps.rock_matrix.perm =[0.000001*milli*darcy*ones(ps.G.cells.
            num,1),0.000001*milli*darcy*ones(ps.G.cells.num,1)
            ,0.000001*milli*darcy*ones(ps.G.cells.num,1)];

        %% Preparing transfer
        aux = ps.dp_info.coefft;
        ps.dp_info.coefft = zeros(ps.G.cells.num,1);
        ps.dp_info.coefft(ps.cell_groups.aquifer_cells) = aux;

        ps.rock_matrix.perm(ps.cell_groups.aquifer_cells,1:2) = ps.
            dp_info.continuum2.kx;
        ps.rock_matrix.perm(ps.cell_groups.aquifer_cells,3) = ps.dp_
            info.continuum2.kz;
        ps.rock_matrix.poro(ps.cell_groups.aquifer_cells) = ps.dp_
            info.continuum2.phi;

        pv_matrix = poreVolume(ps.G, ps.rock_matrix);
end
vb = ps.G.cells.volumes;

%% Define model for compressible fluid
ps.operators.rho = @(p) ps.aquifer.rhow .* exp( ps.aquifer.c * (p -
    0) );

%% Compute transmissibilities
N  = double(ps.G.faces.neighbors);
intInx = all(N ~= 0, 2);
N  = N(intInx, :); % Interior neighbors
hT = computeTrans(ps.G, ps.rock); % Half-transmissibilities

cf = ps.G.cells.faces(:,1);
nf = ps.G.faces.num;
ps.operators.T_all  = 1 ./ accumarray(cf, 1 ./ hT, [nf, 1]);
ps.operators.T  = ps.operators.T_all(intInx);

if(isfield(ps.dp_info,'continuum2'))
        if(strcmp(ps.model_type,'dpdk'))
                hTm = computeTrans(ps.G, ps.rock_matrix); % Half-
                    transmissibilities
                ps.operators.T_all_matrix  = 1 ./ accumarray(cf, 1
                    ./ hTm, [nf, 1]);
```

```
                ps.operators.T_matrix  = ps.operators.T_all_matrix(
                    intInx);
        end
end

%% Define discrete operators
n = size(N,1);
C = sparse( [(1:n)'; (1:n)'], N, ones(n,1)*[-1 1], n, ps.G.cells.num
    );
ps.operators.grad = @(x)C*x;
ps.operators.div  = @(x)-C'*x;
ps.operators.avg  = @(x) 0.5 * (x(N(:,1)) + x(N(:,2)));

%% Boundary Conditions
g = norm(gravity);
ps.cell_groups.bc_faces = find(ps.G.faces.centroids(:,3)<=0.000001);
ps.cell_groups.bc_cells = find(ps.G.cells.centroids(:,3)<=dz);
ps.operators.Tbc = ps.operators.T_all(ps.cell_groups.bc_faces);

%% Define flow equations
gradz  = ps.operators.grad(ps.G.cells.centroids(:,3));
v      = @(p,Trans)  -(Trans/ps.aquifer.muw).*( ps.operators.grad(p)
     - g*ps.operators.avg(ps.operators.rho(p)).*gradz );

ps.equations.presEq = @(p,p0,dt,Is) (pv/dt).*(ps.operators.rho(p)-ps
    .operators.rho(p0))+ps.operators.div(ps.operators.avg(ps.
    operators.rho(p)).*v(p,ps.operators.T) ) - ps.operators.rho(p).*
    Is;

if(isfield(ps.dp_info,'continuum2'))
        if(strcmp(ps.model_type,'dpdk'))
                ps.equations.presEqMatrix = @(pm,pm0,dt) (pv_matrix/
                    dt).*(ps.operators.rho(pm) - ps.operators.rho(pm
                    0))+ps.operators.div(ps.operators.avg(ps.
                    operators.rho(pm)).*v(pm,ps.operators.T_matrix))
                    ;
        else
                ps.equations.presEqMatrix = @(pm,pm0,dt) (pv_matrix/
                    dt).*(ps.operators.rho(pm) - ps.operators.rho(pm
                    0));
        end
```

```
                ps.equations.transferEq = @(pf,pm) ps.aquifer.rhow*vb.*ps.dp
                    _info.coefft.*(pf-pm);
        end

        ps.equations.BCFluxes = @(p,bc_cells,dzg,bc_value,Tbc,muw) (1/muw)*
            Tbc.*ps.operators.rho(p(bc_cells)).*(bc_value - p(bc_cells) +
            dzg);
end

function [] = plotModelScheme(ps,fig)

        G = ps.G;
        marker = zeros(G.cells.num,1);
        marker(ps.cell_groups.aquifer_cells) = 0.5;
        marker(ps.cell_groups.well_cells) = 1;
        figure(fig)
        p = plotCellData(G,marker);
        p.EdgeAlpha = 0;
        view(0,0);
        grid on
        set(gca,'FontSize',16)
        xlabel('x [m]')
        ylabel('Pcap [MPa]')

end

function [pcell] = advancePressure(ps,pcell,dt,Is)

        p1 = pcell{1};
        if(isfield(ps.dp_info,'continuum2'))
                p2 = pcell{2};
        end

        %% Initialize variables
        if(isfield(ps.dp_info,'continuum2'))
                [p1_ad,p2_ad] = initVariablesADI(p1,p2);
        else
                [p1_ad] = initVariablesADI(p1);
        end

        %% Equation numbers
        nc = ps.G.cells.num;
        p1Ix = 1:nc;
```

```
p2Ix = nc+1:2*nc;

%% Main loop
% Newton loop
resNorm = 1e99;
p10  = double(p1_ad); % Previous step pressure
if(isfield(ps.dp_info,'continuum2'))
        p20  = double(p2_ad); % Previous step pressure
end
nit = 0;
tol      = 1e-5; % Newton tolerance
maxits   = 10; % max number of Newton its
while (resNorm > tol) && (nit <= maxits)

        %% Residual of the pressure equation
        eq1 = ps.equations.presEq(p1_ad, p10, dt, Is);

        if(isfield(ps.dp_info,'continuum2'))
                transfer = ps.equations.transferEq(p1_ad,p2_ad);
                eq2 = ps.equations.presEqMatrix(p2_ad, p20, dt)+
                    transfer;
                eq1 = eq1-transfer;
        end

        %% Adding bcs
        bc_value = 0;
        dzg = (ps.G.cells.centroids(ps.cell_groups.bc_cells,3)-
                    ps.G.faces.centroids(ps.cell_groups.bc_faces,3))
            *9.81;
        bcf = ps.equations.BCFluxes(p1_ad,ps.cell_groups.bc_cells,
            dzg,bc_value,ps.operators.Tbc,ps.aquifer.muw);

        eq1(ps.cell_groups.bc_cells) = eq1(ps.cell_groups.bc_cells)
            - bcf;

        %% Concatenating equations
        if(isfield(ps.dp_info,'continuum2'))
                eqs = {eq1, eq2};
                eq  = cat(eqs{:});
        else
                eq = eq1;
        end
```

```
                %% Solving for increment
                J   = eq.jac{1};  % Jacobian
                res = eq.val;     % residual
                upd = -(J \ res); % Newton update
                % Update variables
                p1_ad.val   = p1_ad.val   + upd(p1Ix);

                if(isfield(ps.dp_info,'continuum2'))
                        p2_ad.val   = p2_ad.val   + upd(p2Ix);
                end

                resNorm = norm(res);
                nit     = nit + 1;

        end

        if nit > maxits,
                error('Newton solves did not converge')
        end

        pcell = {p1_ad.val};

        if(isfield(ps.dp_info,'continuum2'))
                pcell{end+1} = p2_ad.val;
        end

end

function [pcell] = integrateTime(ps,pcell,dt,Is,tmax,verbose)

        t = 0;
        while t < tmax,

        %% Advancing time
        t = t + dt;
        [pcell] = ps.advancePressure(pcell,dt,Is);
        if(verbose)
                disp(['Time = ',num2str(t/year)])
        end

        end

end
```

```
end

end
```

Class `PlumeSimulatorDP.m`

```
classdef PlumeSimulatorDP
%% Class PlumeSimulatorDP

properties
        aquifer
        dp_info
        cell_groups
        operators
        equations
        G
end

methods

function [ps] = PlumeSimulatorDP(aquifer,dp_info)
        ps.aquifer = aquifer;
        ps.dp_info = dp_info;
end

function [ps] = setupSimulation(ps,Nx)

        %% Useful variables
        phi1 = ps.dp_info.continuum1.phi;
        swc = ps.aquifer.swc;
        krgmax = ps.aquifer.krnmax;
        muw = ps.aquifer.muw;
        mug = ps.aquifer.mun;
        H = ps.aquifer.H;
        if(isfield(ps.dp_info,'continuum2'))
                phi2 = ps.dp_info.continuum2.phi;
                beta = ps.dp_info.beta;
                Pe = ps.dp_info.Pe;
        end

        %% Set up model: grid
        [nx,ny,nz] = deal( Nx,  1, 1);
```

```
[Dx,Dy,Dz] = deal( ps.aquifer.Lt,  1, 1);
ps.G = cartGrid([nx, ny, nz], [Dx, Dy, Dz]);
ps.G = computeGeometry(ps.G);
dx = Dx/nx;

%% Boundary condition cells
ps.cell_groups.bc_cells = [1,ps.G.cells.num];

%% Computing N
N   = double(ps.G.faces.neighbors);
intInx = all(N ~= 0, 2);
N   = N(intInx, :);

%% Define discrete operators
n    = size(N,1);
C    = sparse( [(1:n)'; (1:n)'], N, ones(n,1)*[-1 1], n, ps.G.cells.
   num);
div  = @(x) -C'*x;
upw  = @(x,flag) x(N(:,1)).*double(flag)+x(N(:,2)).*double(~flag);

mobg = @(hg)(krgmax/mug)*(hg/H);
mobw = @(hg)(1/muw)*(H-hg)/H;
fg = @(hg)(mobg(hg)./(mobg(hg)+mobw(hg)));
fluxf =  @(h,qinj) upw(qinj*fg(h),qinj>0);

%% Define flow equations - Fractured Model
if(isfield(ps.dp_info,'continuum2'))
        transferm = @(h2,h2max,beta)beta*(1-swc)*phi2*(h2max-h2);
        transferf = @(h2,h2max,beta)transferm(h2,h2max,beta);
        ps.equations.h1Eq = @(h1,h10,h2,h2min,h2max,dt,qinj)(dx/dt)*
            phi1*(h1 - h10) + div(fluxf(h1,qinj)) + transferf(h2,h2
            max,beta);

        ps.equations.h2Eq = @(h2,h20,h1,h2min,h2max,dt)(dx/dt)*phi
            2*(1-swc)*(h2 - h20) - transferm(h2,h2max,beta);

        ps.equations.BCFluxLeft = @(h1l,h1l0,h2l,h2minl,h2maxl,dt,
            qinj) phi1*(h1l-h1l0)*dx + qinj*dt *(fg(h1l)-fg(H)) +
            transferf(h2l,h2maxl,beta);

        ps.equations.BCFluxRight = @(h1r,h1rm1,h1r0,h2r,h2minr,h2
            maxr,dt,qinj) phi1*(h1r-h1r0)*dx + qinj*dt*(fg(h1r)-fg(h
            1rm1)) + transferf(h2r,h2maxr,beta);
```

```
        else
                ps.equations.h1Eq   = @(h1,h10,dt,qinj)(dx/dt)*phi1*(1-swc)
                    *(h1 - h10) + div(fluxf(h1,qinj));

                ps.equations.BCFluxLeft = @(h1l,h1l0,dt,qinj)phi1*(1-swc)*(h
                    1l-h1l0)*dx+qinj*dt*(fg(h1l)-fg(H));

                ps.equations.BCFluxRight = @(h1r,h1rm1,h1r0,dt,qinj)phi1*(1-
                    swc)*(h1r-h1r0)*dx+qinj*dt*(fg(h1r)-fg(h1rm1));

        end

end

function [hcell] = advancePlume(ps,hcell,dt,qinj)

        h1 = hcell{1};
        if(isfield(ps.dp_info,'continuum2'))
                h2 = hcell{2};
        end

        %% Useful variables
        delta_rho = (ps.aquifer.rhow-ps.aquifer.rhon);
        g = 9.81;
        if(isfield(ps.dp_info,'continuum2'))
                Pe = ps.dp_info.Pe;
        end

        %% AD variables
        if(isfield(ps.dp_info,'continuum2'))
                [h1_ad,h2_ad] = initVariablesADI(h1,h2);
        else
                h1_ad = initVariablesADI(h1);
        end

        resNorm = 1e99;
        nit = 0;
        tol      = 1e-5; % Newton tolerance
        maxits   = 10;

        %% Equation numbers
        nc = ps.G.cells.num;
```

```
h1Ix = 1:nc;
if(isfield(ps.dp_info,'continuum2'))
        h2Ix = nc+1:2*nc;
end

%% Initial fields
h10 = h1_ad.val;
if(isfield(ps.dp_info,'continuum2'))
        h20 = h2_ad.val;
end

%% Solving plume
while (resNorm > tol) && (nit < maxits)

        if(isfield(ps.dp_info,'continuum2'))
                h2min = Pe/(delta_rho*g);
                h2max = double(h1_ad)-Pe/(delta_rho*g);
                h2max(h2max<0) = 0;

                eq1 = ps.equations.h1Eq(h1_ad,h10,h2_ad,h2min,h2max,
                    dt,qinj);
                eq2 = ps.equations.h2Eq(h2_ad,h20,h1_ad,h2min,h2max,
                    dt);

                % applying bcs
                bc1l = ps.equations.BCFluxLeft(h1_ad(1),h10(1),h2_ad
                    (1),h2min(1),h2max(1),dt,qinj);
                bc1r = ps.equations.BCFluxRight(h1_ad(end),h1_ad(end
                    -1),h10(end),h2_ad(end),h2min(end),h2max(end),dt
                    ,qinj);
                eq1(1) = eq1(1) - bc1l;
                eq1(end) = eq1(end) - bc1r;

                % Concatenating equations
                eqs = {eq1, eq2};
                eq  = cat(eqs{:});
        else
                eq1 = ps.equations.h1Eq(h1_ad,h10,dt,qinj);

                % applying bcs
                bc1l = ps.equations.BCFluxLeft(h1_ad(1),h10(1),dt,
                    qinj);
```

```
                        bc1r = ps.equations.BCFluxRight(h1_ad(end),h1_ad(end
                            -1),h10(end),dt,qinj);
                        eq1(1) = eq1(1) - bc1l;
                        eq1(end) = eq1(end) - bc1r;

                        % Concatenating equations
                        eq = eq1;
                end

                % Solving system
                J   = eq.jac{1};  % Jacobian
                res = eq.val;     % residual
                upd = -(J \ res); % Newton update

                h1_ad.val = h1_ad.val + upd(h1Ix);
                if(isfield(ps.dp_info,'continuum2'))
                        h2_ad.val = h2_ad.val + upd(h2Ix);
                end

                resNorm = norm(res);
                nit     = nit + 1;
        end
        hcell = {h1_ad.val};
        if(isfield(ps.dp_info,'continuum2'))
                hcell{end+1} = h2_ad.val;
                hcell{end+1} = h2max;
        end
end

function [hcell] = integrateTime(ps,hcell,dt,tmax,qinj)

        t = 0;
        while t < tmax

        %% Advancing time
        t = t + dt;
        [hcell] = ps.advancePlume(hcell,dt,qinj);

        end

end

end
```

```
end
```

Class `InjectionRateCalculator.m`

```
classdef InjectionRateCalculator
%% Class InjectionRateCalculator

properties
        pressure_simulator
        injection_rate
    pfinal
end

methods

function [irc] = InjectionRateCalculator(pressure_simulator,Nx,Nz)

        irc.pressure_simulator = pressure_simulator;
        irc.pressure_simulator = irc.pressure_simulator.setupSimulation(Nx,
            Nz);

end

function [irc] = calculateInjectionRate(irc,tmax)

        %% Initial Injection Rate
        Q0 = 50;

        %% Solver options
        options = optimoptions('fsolve');
        options.TolFun = 1*mega*Pascal;

        fun = @(fac)(abs(irc.getMaximumPressureAtCapRock(fac*Q0,tmax)-irc.
            pressure_simulator.aquifer.pfrac));

        fac = fsolve(fun,1,options);

        irc.injection_rate = fac*Q0;

        irc.pfinal = irc.calculateFinalPressureAfterInjection(fac*Q0,tmax);

end
```

```
function [pcell] = calculateFinalPressureAfterInjection(irc,Q,tmax)

        G = irc.pressure_simulator.G;
        aquifer = irc.pressure_simulator.aquifer;
        cell_groups = irc.pressure_simulator.cell_groups;
        ps = irc.pressure_simulator;
        dt = 1*year;

        g = 9.81;
        p_init = aquifer.rhow * g * G.cells.centroids(:,3);

        pf = p_init;
        pm = p_init;
        pcell = {pf,pm};

        Is = zeros(G.cells.num,1);
        Is(cell_groups.well_cells) = Q/aquifer.W/length(cell_groups.well_
            cells);
        [pcell] = ps.integrateTime(pcell,dt,Is,tmax,0);

end

function [pmax] = getMaximumPressureAtCapRock(irc,Q,tmax)

        cell_groups = irc.pressure_simulator.cell_groups;
    p = calculateFinalPressureAfterInjection(irc,Q,tmax);
    pcap = p{1}(cell_groups.cap_rock_cells);
    pmax = max(pcap);

end

end

end
```

Madison (a)

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	600	m
D	Szulczewski et al. (2012)	3000	m
B	Szulczewski et al. (2012)	2000	m
W	Szulczewski et al. (2012)	100	km
L	Szulczewski et al. (2012)	200	km
L_t	Szulczewski et al. (2012)	60	km
T_s	Szulczewski et al. (2012)	6	^oC
G_T	Szulczewski et al. (2012)	40	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	30	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	60	mD
k_z^{aq}	Szulczewski et al. (2012)	60	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.08	-
P_{frac}	Szulczewski et al. (2012)	30	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.4	cP
μ_n	Szulczewski et al. (2012)	0.04	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	500	kg/m^3
P_{res}	calculated (hydrostatic of brine)	29.5	MPa
T_{res}	calculated (linear geothermal profile)	126	^oC
β	calculated (Equation (3.9))	3.2×10^{-8}	s^{-1}
P_e	calculated (Equation (5.20))	58.4	kPa

Mt. Simon (a) Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	400	m
D	Szulczewski et al. (2012)	2000	m
B	Szulczewski et al. (2012)	0	m
W	Szulczewski et al. (2012)	200	km
L	Szulczewski et al. (2012)	200	km
L_t	Szulczewski et al. (2012)	100	km
T_s	Szulczewski et al. (2012)	10	^oC
G_T	Szulczewski et al. (2012)	20	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	50	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	100	mD
k_z^{aq}	Szulczewski et al. (2012)	100	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	40	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.8	cP
μ_n	Szulczewski et al. (2012)	0.06	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	700	kg/m^3
P_{res}	calculated (hydrostatic of brine)	19.7	MPa
T_{res}	calculated (linear geothermal profile)	50	^oC
β	calculated (Equation (3.9))	9.94×10^{-9}	s^{-1}
P_e	calculated (Equation (5.20))	82.5	kPa

Mt. Simon (c) Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	100	m
D	Szulczewski et al. (2012)	1000	m
B	Szulczewski et al. (2012)	0	m
W	Szulczewski et al. (2012)	200	km
L	Szulczewski et al. (2012)	200	km
L_t	Szulczewski et al. (2012)	300	km
T_s	Szulczewski et al. (2012)	15.2	^oC
G_T	Szulczewski et al. (2012)	20	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	50	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	100	mD
k_z^{aq}	Szulczewski et al. (2012)	100	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	20	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	1	cP
μ_n	Szulczewski et al. (2012)	0.06	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	700	kg/m^3
P_{res}	calculated (hydrostatic of brine)	9.9	MPa
T_{res}	calculated (linear geothermal profile)	35.2	^oC
β	calculated (Equation (3.9))	3.47×10^{-8}	s^{-1}
P_e	calculated (Equation (5.20))	90.3	kPa

Morrison Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	200	m
D	Szulczewski et al. (2012)	2000	m
B	Szulczewski et al. (2012)	1000	m
W	Szulczewski et al. (2012)	100	km
L	Szulczewski et al. (2012)	100	km
L_t	Szulczewski et al. (2012)	100	km
T_s	Szulczewski et al. (2012)	10	^oC
G_T	Szulczewski et al. (2012)	30	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	40	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	70	mD
k_z^{aq}	Szulczewski et al. (2012)	70	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	50	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.4	cP
μ_n	Szulczewski et al. (2012)	0.04	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	600	kg/m^3
P_{res}	calculated (hydrostatic of brine)	19.7	MPa
T_{res}	calculated (linear geothermal profile)	70	^oC
β	calculated (Equation (3.9))	3.66×10^{-8}	s^{-1}
P_e	calculated (Equation (5.20))	93.11	kPa

Cedar Keys Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	400	m
D	Szulczewski et al. (2012)	2000	m
B	Szulczewski et al. (2012)	2000	m
W	Szulczewski et al. (2012)	200	km
L	Szulczewski et al. (2012)	100	km
L_t	Szulczewski et al. (2012)	100	km
T_s	Szulczewski et al. (2012)	20	^oC
G_T	Szulczewski et al. (2012)	20	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	5	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	10	mD
k_z^{aq}	Szulczewski et al. (2012)	10	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	30	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.7	cP
μ_n	Szulczewski et al. (2012)	0.07	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	800	kg/m^3
P_{res}	calculated (hydrostatic of brine)	19.7	MPa
T_{res}	calculated (linear geothermal profile)	60	^oC
β	calculated (Equation (3.9))	7.89×10^{-10}	s^{-1}
P_e	calculated (Equation (5.20))	253.5	kPa

St. Peter Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	40	m
D	Szulczewski et al. (2012)	2000	m
B	Szulczewski et al. (2012)	1000	m
W	Szulczewski et al. (2012)	100	km
L	Szulczewski et al. (2012)	400	km
L_t	Szulczewski et al. (2012)	100	km
T_s	Szulczewski et al. (2012)	10	^oC
G_T	Szulczewski et al. (2012)	20	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	30	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	50	mD
k_z^{aq}	Szulczewski et al. (2012)	50	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.06	-
P_{frac}	Szulczewski et al. (2012)	30	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	1	cP
μ_n	Szulczewski et al. (2012)	0.07	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	800	kg/m^3
P_{res}	calculated (hydrostatic of brine)	19.7	MPa
T_{res}	calculated (linear geothermal profile)	50	^oC
β	calculated (Equation (3.9))	1.61×10^{-7}	s^{-1}
P_e	calculated (Equation (5.20))	63.9	kPa

Potomac Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	400	m
D	Szulczewski et al. (2012)	1000	m
B	Szulczewski et al. (2012)	0	m
W	Szulczewski et al. (2012)	100	km
L	Szulczewski et al. (2012)	100	km
L_t	Szulczewski et al. (2012)	40	km
T_s	Szulczewski et al. (2012)	10	^oC
G_T	Szulczewski et al. (2012)	30	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	2000	mD
k_z^{cap}	Szulczewski et al. (2012)	0.2	mD
k_x^{aq}	Szulczewski et al. (2012)	3000	mD
k_z^{aq}	Szulczewski et al. (2012)	3000	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	20	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.7	cP
μ_n	Szulczewski et al. (2012)	0.04	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	500	kg/m^3
P_{res}	calculated (hydrostatic of brine)	9.9	MPa
T_{res}	calculated (linear geothermal profile)	40	^oC
β	calculated (Equation (3.9))	5.73×10^{-7}	s^{-1}
P_e	calculated (Equation (5.20))	16.9	kPa

Navajo-Nugget (a) Aquifer

Property	Source	Value	Unit
L_z	Szulczewski et al. (2012)	200	m
D	Szulczewski et al. (2012)	3000	m
B	Szulczewski et al. (2012)	500	m
W	Szulczewski et al. (2012)	200	km
L	Szulczewski et al. (2012)	90	km
L_t	Szulczewski et al. (2012)	30	km
T_s	Szulczewski et al. (2012)	6	^oC
G_T	Szulczewski et al. (2012)	30	$^oC/km$
k_x^{cap}	Szulczewski et al. (2012)	50	mD
k_z^{cap}	Szulczewski et al. (2012)	0.02	mD
k_x^{aq}	Szulczewski et al. (2012)	100	mD
k_z^{aq}	Szulczewski et al. (2012)	100	mD
ϕ^{aq}	Szulczewski et al. (2012)	0.2	-
P_{frac}	Szulczewski et al. (2012)	70	MPa
c	Szulczewski et al. (2012)	0.1	GPa^{-1}
μ_w	Szulczewski et al. (2012)	0.4	cP
μ_n	Szulczewski et al. (2012)	0.05	cP
k_{rn}^{max}	Szulczewski et al. (2012)	0.6	-
S_{wr}	Szulczewski et al. (2012)	0.4	-
ρ_w	Szulczewski et al. (2012)	1000	kg/m^3
ρ_n	Szulczewski et al. (2012)	600	kg/m^3
P_{res}	calculated (hydrostatic of brine)	29.5	MPa
T_{res}	calculated (linear geothermal profile)	96	^oC
β	calculated (Equation (3.9))	5.11×10^{-8}	s^{-1}
P_e	calculated (Equation (5.20))	72.6	kPa

REFERENCES

S. A. A. Abushaikha and O. R. Gosselin. Matrix-Fracture Transfer Function in Dual-Medium Flow Simulation: Review, Comparison, and Validation. 2008.

S. Agada, S. Geiger, and F. Doster. Wettability, hysteresis and fracture-matrix interaction during CO2 EOR and storage in fractured carbonate reservoirs. *International Journal of Greenhouse Gas Control*, 46:57–75, 2016.

M. Ahmed Elfeel, A. Al-Dhahli, S. Geiger, and M. I. J. van Dijke. Fracture-matrix interactions during immiscible three-phase flow. *Journal of Petroleum Science and Engineering*, 143:171–186, 2016.

Y. Al-Khulaifi, Q. Lin, M. Blunt, and B. Bijeljic. Reservoir-condition pore-scale imaging of dolomite reaction with supercritical CO2 acidified brine: Effect of pore-structure on reaction rate using velocity distribution analysis. *International Journal of Greenhouse Gas Control*, 68:99–111, 2018.

A. Alhammadi, A. Alratrout, B. Bijeljic, and M. Blunt. In situ Wettability Measurement in a Carbonate Reservoir Rock at High Temperature and Pressure. *SPE Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, 13-16 November 2017*, 2017.

T. Arbogast. A simplified dual-porosity model for 2-phase flow. *Computational methods in water resources IX. Mathematical modeling in water resources*, 2:419–426, 1992.

J. S. Aronofsky, L. Masse, and S. G. Natanson. A Model for the Mechanism of Oil Recovery from the Porous Matrix Due to Water Invasion in Fractured Reservoirs. *SPE Journal*, 213(2):2–4, 1958.

E. Aspenes, G. Ersland, A. Graue, J. Stevens, and B. Baldwin. Wetting Phase Bridges Establish Capillary Continuity Across Open Fractures and Increase Oil Recovery in Mixed-Wet Fractured Chalk. *Transport in Porous Media*, 74: 35–47, 2008.

N. Barton, S. Bandis, and K. Bakhtar. Strength, deformation and conductivity copuling of rock joints. *International Journal of Rock Mechanics and Mining Sciences*, 22(3):121–140, 1985.

J. Bear. *Dynamics of Fluids in Porous Media*. American Elsevier Publishing Company, 1972.

N. Bech, O.K. Jensen, and B. Nielsen. Modeling of Gravity-Imbibition and Gravity-Drainage Processes. *SPE*, (February):129–136, 1991.

B. L. Beckner, H. M. Chan, A. E. McDonald, S. O. Wooten, and Jones. Simulating Naturally Fractured Reservoirs Using a Subdomain Method. *11th SPE Symposium on Reservoir Simulation*, 1991.

B. Bennion and S. Bachu. Drainage and Imbibition CO2/Brine Relative Permeability Curves at Reservoir Conditions for Carbonate Formations. *SPE Journal*, pages 1–18, 2010.

D. B. Bennion and S. Bachu. Supercritical CO2 and H2S-Brine Drainage and Imbibition Relative Permeability Relationships for Intergranular Sandstone and Carbonate Formations. *SPE*, 2006.

T. I. Bjornara and S. Mathias. A pseudospectral approach to the McWhorter and Sunada equation for two-phase flow in porous media with capillary pressure. *Computational Geosciences*, 17(6):889–897, July 2013.

B. J. Bourbiaux, R. Basquet, M. Cacas, J. Daniel, and S. Sarda. An Integrated Workflow to Account for Multi-Scale Fractures in Reservoir Simulation Models : Implementation and Benefits. *SPE Journal*, 2002.

I. C. Bourg, L. E Beckingham, and D. J. DePaolo. The Nanoscale Basis of CO2 Trapping for Geologic Storage. *Environmental science & technology*, 49:10265–10284, aug 2015.

S. E. Buckley and M. C. Leverett. Mechanisms of Fluid Displacement in Sands. *Transactions of the American Institute of Mining, Metallurgical and Petroleum Engineers*, (146):107–116, 1942.

T. P. Burchette. Carbonate rocks and pertroleum reservoirs: a geological perspectiive from the industry. *Advances in Carbonate Exploration and Reservoir Analysis. Geological Society, London, Special Publications*, 370:17–37, 2012.

A. Cavanagh and P. Ringrose. Simulation of CO2 distribution at the In Salah storage site using high-resolution field-scale models. *Energy Procedia*, 4:3730–3737, 2011. doi: 10.1016/j.egypro.2011.02.306.

M. A. Celia, S. Bachu, J. M. Nordbotten, and K. W. Bandilla. Status of CO2 storage in deep saline aquifers with emphasis on modeling approaches and practical simulations. *Water Resources Research*, 51(9):6846–6892, 2015. doi: 10.1002/2015WR017609.

J. Chen, M. A. Miller, and K. Sepehrnoori. Theoretical Investigation of Countercurrent Imbibition in Fractured Reservoir Matrix Blocks. *SPE Journal*, 1995.

P. Chiquet, J. Daridon, D. Broseta, and S. Thibeau. CO2/water interfacial tensions under pressure and temperature conditions of CO2 geological storage. *Energy Conversion and Management*, 48:736–744, 2007.

K. Coats. Implicit Compositional Simulation of Single-Porosity and Dual-Porosity Reservoirs (SPE-18427). *SPE Journal*, 1989.

Computer Modelling Group. IMEX User Guide. 2016.

R. M. Cuellar-Franca and A. Azapagic. Carbon capture, storage and utilisation technologies: A critical analysis and comparison of their life cycle environmental impacts. *Journal of CO2 Utilization*, 9:82–102, 2015. doi: 10.1016/j.jcou.2014.12.001.

M. R. Dernaika, M. A. Basioni, A. Dawoud, and M. Z. Kalam. Variations in Bounding and Scanning Relative Permeability Curves With Different Carbonate Rock Types. *SPE Journal*, pages 11–14, 2013.

G. Di Donato, Z. Tavassoli, and M. J. Blunt. Analytical and numerical analysis of oil recovery by gravity drainage. *Journal of Petroleum Science and Engineering*, 54(1-2):55–69, nov 2006.

G. Di Donato, H. Lu, Z. Tavassoli, and M. J. Blunt. Multirate-Transfer Dual-Porosity Moeling of Gravity Drainage and Imbibition. *SPE Journal*, March: 77–88, 2007.

F. Doster, O. Hönig, and R. Hilfer. Horizontal flow and capillarity-driven redistribution in porous media. *Physical Review E*, 86(1):16317, 2012.

D. O. Egya, S. Geiger, P. W. M. Corbett, R. March, K. Bisdom, G. Bertotti, and F. H. Bezerra. Analysing the limitations of the dual-porosity response during well tests in naturally fractured reservoirs. *Petroleum Geoscience*, 2018. doi: 10.1144/petgeo2017-053.

O. Eiken, P. Ringrose, C. Hermanrud, B. Nazarian, T. A. Torp, and Lars Hoier. Lessons Learned from 14 years of CCS Operations: Sleipner, In Salah and Snohvit. *Energy Procedia*, 4:5541–5548, 2011.

H. Emami-Meybodi and H. Hassanzadeh. Advances in Water Resources Two-phase convective mixing under a buoyant plume of CO2 in deep saline aquifers. *Advances in Water Resources*, 76:55–71, 2015.

M. A. Fernø, Å. Haugen, S. Wickramathilaka, J. Howard, A. Graue, G. Mason, and N. R. Morrow. Magnetic resonance imaging of the development of fronts during spontaneous imbibition. *Journal of Petroleum Science and Engineering*, 101:1–11, jan 2013.

A. Firoozabadi and L. K. Thomas. Sixth SPE Comparative Solution Project: Dual-Porosity Simulators. *Journal of Petroleum Technology*, 42(06):710–763, 1990.

A. Fumagalli, L. Pasquale, S. Zonca, and S. Micheletti. An Upscaling Procedure for Fractured Reservoirs with Embedded Grids. *Water Resources Research*, 52: 6506–6525, 2016. doi: 10.1029/2008WR006912.M.

S. Geiger, S. K. Matthai, J. Niessner, and R. Helmig. Black-Oil Simulations for Three-Component, Three-Phase Flow in Fractured Porous Media. *SPE Journal*, (June), 2009.

S. Geiger, M. Dentz, and I. Neuweiler. A Novel Multirate Dual-Porosity Model for Improved Simulation of Fractured and Multiporosity Reservoirs. *SPE Journal*, August:670–684, 2013.

J.R. Gilman. An Efficient Finite-Difference Method for Simulating Phase Segregation in the Matrix Blocks in Double-Porosity Reservoirs. *SPE Reservoir Engineering*, 1(4):403–413, 1986.

B. Gong, M. Karimi-Fard, and L. Durlofsky. Upscaling Discrete Fracture Characterizations to Dual-Porosity, Dual-Permeability Models for Efficient Simulation of Flow With Strong Gravitational Effects. *SPE Journal*, March, 2008.

B. Guo, K. W. Bandilla, F. Doster, E. Keilegavlen, and M. A. Celia. A vertically integrated model with vertical dynamics for CO2 storage. *Water Resources Research*, 50(August), 2014. doi: 10.1002/2013WR015215.

J. Hagoort. Oil Recovery by Gravity Drainage. *SPE*, (June):139–150, 1980.

H. Hoteit and A. Firoozabadi. Multicomponent fluid flow by discontinuous Galerkin and mixed methods in unfractured and fractured media. *Water Resources Research*, 41(11), nov 2005.

R. Huber and Helmig R. Multiphase flow in heterogeneous porous media: A classical finite element method versus an implicit pressure-explicit saturation-based mixed finite element-finite volume approach. *International Journal for Numerical Methods in Fluids*, 29(8):899–920, 1999.

M. Hui, M. Karimi-Fard, B. Mallison, and L. Durlofsky. A General Modeling Framework for Simulating Complex Recovery Processes in Fractured Reservoirs at Different Resolutions. *SPE Journal*, 2018.

M. Iding and P. Ringrose. Evaluating the impact of fractures on the performance of the In Salah CO2 storage site. *International Journal of Greenhouse Gas Control*, 4(2):242–248, 2010.

IPCC. Carbon Dioxide Capture and Storage. Technical report, 2005.

R. Juanes and R. MacMinn. Upscaling of capillary trapping under gravity override: Application to CO2 sequestration in aquifers. *SPE Symposium on Improved Oil Recovery*, 2008.

R. Juanes, E. J. Spiteri, F. M. Orr, and M. J. Blunt. Impact of relative permeability hysteresis on geological CO2 storage. *Water Resources Research*, 42(12), December 2006.

M. Karimi-Fard, L. J. Durlofsky, and K. Aziz. An Efficient Discrete-Fracture Model Applicable for General-Purpose Reservoir Simulators. *SPE Journal*, (February 2003):3–5, 2004.

M. Karimi-Fard, B. Gong, and L. Durlofsky. Generation of coarse-scale continuum flow models from detailed fracture characterizations. *Water Resources Research*, 42(10), 2006.

H. Kazemi, L. S. Merrill Jr, K. L Porterfield, and P. R. Zeman. Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs. *SPE Journal*, pages 317–326, 1976.

H. Kazemi, J. R Gilman, and a. M Eisharkawy. Analytical and Numerical Solution of Oil Recovery From Fractured Reservoirs With Empirical Transfer Functions. *SPE Journal*, (May):219–227, 1992.

S. Krogstad, K. Lie, O. Møyner, H. Nilsen, X. Raynaud, and B. Skaflestad. MRST-AD - an Open-Source Framework for Rapid Prototyping and Evaluation of Reservoir Simulation Problems. *SPE Reservoir Simulation Symposium*, 2015.

A. Labastie. Capillary continuity between blocks of a fractured reservoir. *65th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, New Orleans, USA*, 1990.

D. Law and S. Bachu. Hydrogeological and numerical analysis of CO2 disposal in deep aquifers in the Alberta sedimentary basin. *Energy Conversion and Management*, 37(6-8):1167–1174, 1996.

S. Lee, M. Lough, and C. Jensen. Hierarchical Modeling of Flow in Naturally Fractured Formations with Multiple Length Scales. *Water Resources Research*, 37(3):443–455, 2001.

P. Lemonnier and Bernard J. Bourbiaux. Simulation of Naturally Fractured Reservoirs . State of the Art - Part 2: Matrix-Fracture Transfers and Typical Features of Numerical Studies. *Oil & Gas Science and Technology Revue de l'Institut Francais du Petrole*, 65(2):263–286, apr 2010.

K. Li and R. N. Horne. Generalized Scaling Approach for Spontaneous Imbibition : An Analytical Model. *SPE Journal*, 9(3):251–258, 2006.

K. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322, aug 2011.

K. Lim and K. Aziz. Matrix-fracture transfer shape factors for dual-porosity simulators. *Journal of Petroleum Science and Engineering*, 13:169–178, 1995.

H. Lu, G. Di Donato, and M. J. Blunt. General Transfer Functions for Multiphase Flow in Fractured Reservoirs. *SPE Journal*, pages 24–27, 2008.

R. Lucas. Über das Zeitgesetz des kapillaren Aufstiegs von Flüssigkeiten. *Colloid Polymer Science*, 23(1):15–22, 1918.

S. Ma, N. R. Morrow, and X. Zhang. Generalized Scaling of Spontaneous Imbibition Data for Strongly Water-Wet Systems. *Journal of Petroleum Science and Engineering*, 4105(18):165–178, 1997.

S. Ma, X. Zhang, and N. Morrow. Influence of Fluid Viscosity on Mass Transfer Between Rock Matrix and Fractures. *Journal of Canadian Petroleum Technology*, 38(7):25–30, 1999.

C. W. Macminn, M. L. Szulczewski, and R. Juanes. CO2 migration in saline aquifers. Part 2 - Capillary and solubility trapping. *Journal of Fluid Mechanics*, pages 321–351, 2011. doi: 10.1017/jfm.2011.379.

C. Maier and S. Geiger. Multi-rate mass-transfer dual-porosity modelling using the exact analytical solution for spontaneous imbibition. *75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013*, pages 10–13, 2013.

R. March, F. Doster, and S. Geiger. Accurate Early-Time and Late-Time Modeling of Countercurrent Spontaneous Imbibition. *Water Resources Research*, 52:6263–6276, 2016. doi: 10.1029/2008WR006912.M.

Mathworks. *Matlab User's Guide r2014b*. 2014.

C. C. Mattax and J. R. Kyte. Imbibition Oil Recovery from Fractured, Water-Drive Reservoir. *SPE Journal*, 2(2):177–184, 1962.

D. B. McWhorter and D. K. Sunada. Exact Integral Solutions for Two-Phase Flow. *Water Resources Research*, 26(3):399–413, 1990.

H. Menke, C. Reynolds, M. Andrew, J. Pereira Nunes, B. Bijeljic, and M. Blunt. 4D multi-scale imaging of reactive flow in carbonates: Assessing the impact of heterogeneity on dissolution regimes using streamlines at multiple length

scales Multiscale imaging Reactive transport Core scale Pore scale X-ray microtomography CO2 storage. *Chemical Geology*, 481:27–37, 2018.

A. Moinfar, A. Varavei, K. Sepehrnoori, and R. T. Johns. Development of a Coupled Dual Continuum and Discrete Fracture Model for the Simulation of Unconventional Reservoirs. *SPE Reservoir Simulation Symposium*, pages 1–17, feb 2013.

A. Moinfar, A. Varavei, and K. Sepehrnoori. Development of an Efficient Embedded Discrete Fracture Model for 3D Compositional Reservoir Simulation in Fractured Reservoirs. *SPE Journal*, (April):289–303, 2014.

W. Narr, D. Schechter, and L. Thompson. *Naturally Fractured Reservoir Characterization*. Society of Petroleum Engineers, 2006.

J. M. Nordbotten and M. A. Celia. *Geological Storage of CO2: Modeling Approaches for Large-Scale Simulation*. Wiley, 2012. ISBN 978-0-470-88946-6.

S. Pacala and R. Socolow. Stabilization Wedges: Solving the Climate Problem for the Next 50 Years with Current Technologies. *Science*, 305(5686):968–972, 2004.

Y. Pamukcu, S. Hurter, L. Jammes, D. Vu-Hoang, and L. Pekot. Characterizing and predicting short term performance for the in Salah Krechba field CCS joint industry project. *Energy Procedia*, 4:3371–3378, 2011.

B. Pedrera, H. Bertin, U. Bordeaux, G. Hamon, and A. Augustin. Wettability Effect on Oil Relative Permeability During a Gravity Drainage. *SPE Journal*, 77542, 2002.

K. Pruess. A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media. *Society of Petroleum Engineers Journal*, 25(01):14–26, apr 1985.

P. Quandalle and J. C. Sabathier. Typical Features of a Multiphase Reservoir Simulator. *SPE Journal*, (November):475–480, 1989.

B. Ramirez, H. Kazemi, and M. Al-kobaisi. A Critical Review for Proper Use of Water/Oil/Gas Transfer Functions in Dual - Porosity Naturally Fractured Reservoirs: Part I. *SPE Reservoir Evaluation & Engineering*, 12(02):200 – 210, 2009.

A. Riaz, M. Hesse, H. A. Tchelepi, and F. M. Orr. Onset of convection in a gravitationally unstable diffusive boundary layer in porous media. *Journal of Fluid Mechanics*, 548(-1):87, feb 2006.

P. S. Ringrose. Principles of sustainability and physics as a basis for the low-carbon energy transition. *Petroleum Geoscience*, 2017.

W. R. Rossen and A.T. Kumar. Single- and Two-Phase Flow in Natural Fractures. *SPE Annual Technical Conference and Exhibition, Washington DC, USA. SPE 24915*, 1992.

J. Rutqvist. The geomechanics of CO2 storage in deep sedimentary formations. *Geotechnical and Geological Engineering*, 30(3):525–551, 2012.

J. Rutqvist, D. Vasco, and L. Myer. Coupled reservoir-geomechanical analysis of CO2 injection and ground deformations at In Salah, Algeria. *International Journal of Greenhouse Gas Control*, 4(2):225–230, 2010.

S. Saraji, L. Goual, M. Piri, and H. Plancher. Wettability of Supercritical Carbon Dioxide/Water/Quartz Systems: Simultaneous Measurement of Contact Angle and Interfacial Tension at Reservoir Conditions. *Langmuir*, 29:6856–6866, 2013.

Schlumberger. Eclipse Technical Description. Technical report, 2014.

K. S. Schmid and S. Geiger. Universal scaling of spontaneous imbibition for water-wet systems. *Water Resources Research*, 48(3), mar 2012. doi: 10.1029/2011WR011566.

K. S. Schmid and S. Geiger. Universal scaling of spontaneous imbibition for arbitrary petrophysical properties: Water-wet and mixed-wet states and Handy's conjecture. *Journal of Petroleum Science and Engineering*, 101:44–61, January 2013.

K. S. Schmid, S. Geiger, and K. S. Sorbie. Semianalytical solutions for cocurrent and countercurrent imbibition and dispersion of solutes in immiscible two-phase flow. *Water Resources Research*, 47(2), feb 2011. doi: 10.1029/2010WR009686.

K. S. Schmid, S. Geiger, and K. S. Sorbie. Higher order FE-FV method on unstructured grids for transport and two-phase flow with variable viscosity in heterogeneous porous media. *Journal of Computational Physics*, 241:416–444, may 2013.

S. Shah, O. Møyner, M. Tene, K. Lie, and H. Hajibeygi. The multiscale restriction smoothed basis method for fractured porous media (F-MsRSB). *Journal of Computational Physics*, 318:36–57, 2016.

F. Sonier, P. Souillard, and F. T. Blaskovich. Numerical Simulation of Naturally Fractured Reservoirs. *SPE Journal*, 3(04):1114–1122, 1988.

N. Spycher, K. Pruess, and J. Ennis-King. CO2-H2O mixtures in the geological sequestration of CO2. I. Assessment and calculation of mutual solubilities

from 12 to 100Â°C and up to 600 bar. *Geochimica et Cosmochimica Acta*, 67(16): 3015–3031, 2003.

T. Suekane, T. Nobuso, S. Hirai, and M. Kiyota. Geological storage of carbon dioxide by residual gas and solubility trapping. *International Journal of Greenhouse Gas Control*, 2(1):58–64, 2008. doi: 10.1016/S1750-5836(07)00096-5.

M. Szulczewski, C. MacMinn, H. Herzog, and R. Juanes. Lifetime of carbon capture and storage as a climate-change mitigation technology. *Proceedings of the National Academy of Sciences USA*, 109(14):5185–5189, 2012.

S. Taku Ide, K. Jessen, and F. M. Orr. Storage of CO2 in saline aquifers: Effects of gravity, viscous, and capillary forces on amount and timing of trapping. *International Journal of Greenhouse Gas Control*, 1(4):481–491, October 2007.

Z. Tavassoli, R. W. Zimmerman, and M. J. Blunt. Analytic Analysis for Oil Recovery During Counter-Current Imbibition in Strongly Water-Wet Systems. *Transport in Porous Media*, 58(1-2):173–189, January 2005.

J. Tecklenburg, I. Neuweiler, M. Dentz, J. Carrera, S. Geiger, C. Abramowski, and O. Silva. A non-local two-phase flow model for immiscible displacement in highly heterogeneous porous media and its parametrization. *Advances in Water Resources*, 62:475–487, December 2013.

J. Tecklenburg, I. Neuweiler, J. Carrera, and M. Dentz. Multi-rate mass transfer modeling of two-phase flow in highly heterogeneous fractured and porous media. *Advances in Water Resources*, 91:63–77, 2016.

L. Thomas, T. Dixon, and R. Pierson. Fractured Reservoir Simulation. *SPE Journal*, 23(1):42–54, 1983.

J. Verdon, J. Kendall, A. Stork, R. Chadwick, D. White, and R. Bissell. Comparison of geomechanical deformation induced by megatonne-scale CO2 storage at Sleipner, Weyburn, and In Salah. *Proceedings of the National Academy of Sciences USA*, 110(30):E2762–71, 2013.

V. Vilarrasa, D. Bolster, M. Dentz, S. Olivella, and J. Carrera. Effects of CO2 Compressibility on CO2 Storage in Deep Saline Aquifers. *Transport in Porous Media*, 85(2):619–639, 2010.

K. Vujevic, T. Graf, C. Simmons, and A. Werner. Impact of fracture network geometry on free convective flow patterns. *Advances in Water Resources*, 71: 65–80, 2014.

J. E. Warren and P. J. Root. The Behavior of Naturally Fractured Reservoirs. *SPE Journal*, September:245–255, 1963.

E. W. Washburn. The Dynamics of Capillary Flow. *The Physical Review*, XVII(3): 273–283, 1921.

D. Zhou, L. Jia, J. Kamath, and A. R. Kovscek. Scaling of counter-current imbibition processes in low-permeability porous media. *Journal of Petroleum Science and Engineering*, 33(1-3):61–74, April 2002.

Q. Zhou, C. M. Oldenburg, L. H. Spangler, and J. T. Birkholzer. Approximate Solutions for Diffusive Fracture-Matrix Transfer: Application to Storage of Dissolved CO2 in Fractured Rocks. *Water Resources Research*, 53:1746–1762, 2014. doi: 10.1002/2013WR014979.Reply.

R. W. Zimmerman, G. Chen, T. Hadgu, and G. S. Bodvarsson. A numerical dual-porosity model with semianalytical treatment of fracture/matrix flow. *Water Resources Research*, 29(7):2127–2137, 1993. doi: 10.1029/93WR00749.

R. W. Zimmerman, T. Hadgut, and G. S. Bodvarsson. A new lumped-parameter model for flow in unsaturated dual-porosity media. *Advances in Water Resources*, 19(5):317–327, 1996.